1-1-2019

Problems in Control, Estimation, and Learning in Complex Robotic Systems

Seyyedhossein Mousavi
Lehigh University

Problems in Control, Estimation, and Learning in Complex Robotic Systems

by

Hossein K. Mousavi

(Officially, SeyyedHossein Mousavi)

Presented to the Graduate and Research Committee of

Lehigh University

in Candidacy for the Degree of

Doctor of Philosophy

in

Mechanical Engineering

Lehigh University

January 2020

Approved and recommended for acceptance as a dissertation in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

_____
Date

_____
Dissertation Advisor

Committee Members:

_____
Prof. Nader Motee, Committee Chair

_____
Prof. Mehran Mesbahi

_____
Prof. Martin Takáč

_____
Prof. Subhrajit Bhattacharya

To my lovely girlfriend, Elnaz...

and to my mother, and my late father...

Acknowledgements

First and foremost, I would like to genuinely thank my advisor, Prof. Nader Motee for being a great mentor. Through his great ideas, he introduced me to research in systems and control theory. He made me understand the beauty of applied mathematics and how a great theory can subtly explain a huge range of topics at once. During this time, he persuaded me to always go after new ideas and topics and to try solving problems that require learning new ideas. He has been a supportive and passionate mentor and I am thankful for that. To be honest, at first, I did not have any clues of how my final dissertation would look like. Nevertheless, I trusted Prof. Motee with his leadership. I do not hesitate to say that, overall, he guided me through a Ph.D. experience that I am proud of. I would like to extend my sincere gratitude to Prof. Mehran Mesbahi, Prof. Subhrajit Bhattacharya, and Prof. Martin Takáč for accepting my request to be on my dissertation committee. I am indebted for their time and consideration by attending my general exam, Ph.D. proposal defense, and Ph.D. dissertation defense, despite their busy schedules. I have tremendously enjoyed Prof. Bhattacharya's course on advanced dynamics and vibration. My first interaction with Prof. Takáč was in OptML meetings in 2016. Later on, I attended his course on Mining Large datasets. Initially, I got introduced to the notions of machine learning and data through his valuable course. The last two chapters of this dissertation would not be possible without his collaboration and support. His subtle sense of humor coupled with his humility and intelligence made working with his tremendously joyful. I should thank two wonderful friends of mine that had been working at the Department of Mechanical Engineering and Mechanics: Ms. Jennifer Smith and Ms. Allison Marsteller. Jenn just retired from Lehigh. She always went out of her way to help. Her positive energy was the thing that I remember

from every single conversation. Ali has been the kind graduate coordinator of the MEM department since 2016. She always welcomes you with her wide smile in her office and makes sure you have every bit of needed information. I should also thank my B.Sc. advisor Sharif University of Technology, Prof. Amir Jalali. His passion for research in dynamical systems made me very interested in starting my Ph.D. in systems theory. Additionally, I am grateful for Prof. Aria Alasty. Through his introductory and advanced control systems courses, I got amazed by the power of control theory. Next, I would like to thank my collaborators Dr. MirSaleh Bahavarnia, Dr. Christoforos Somarakis, Dr. Mohammadreza Nazari, and Prof. Qiyu Sun. MirSaleh is a true friend and smart scholar. His vast knowledge of linear control theory and linear algebra made him my *gold standard* reference in these topics. Chris was a postdoctoral scholar and then a research scientist in our lab. I learned a lot from collaborating with him. His passionate *Greek* way of describing his ideas made him a unique source of inspiration in our lab. Working with Reza on reinforcement learning problems was a joyful quest and I learned a lot from him during our collaboration. I should also thank Mr. Arash Amini. Arash is smart, kind, and open to conversation. I have enjoyed our long conversations about almost anything in the time that he was in our lab. He is not afraid of changing his mind on any given topic. I should thank my true friend Amin Ghafari for his valuable almost-decade-long friendship. There is so much to say about him, but I can put it this way,"*everything is better off if it's new unless it's a friend!*", as we say in Persian. I would like to thank my family: my Mother, Malakeh, and my sweet sisters: Narges, Nasrin, and Nafiseh. During this time, I had lost my father, who gracefully lost the battle to cancer, and they were there for me trying to cope with the pain that is still sitting on top of my heart today. Last but not least, I am very much indebted to the love of my life, my girlfriend, Elnaz. Without her support, finishing this dissertation would have been impossible. I would like to try and thank her as much as possible, however, her unconditional love has persuaded me that words can never be adequate. She is the most wonderful person that I have ever met. While at times during at my Ph.D., I have been borderline giving up, she was /the one pushing me to get back on track. She was also the one who was there for me when I lost my father away from home.

Contents

List of Tables

List of Figures

Abstract

In this dissertation, we consider a range of different problems in systems, control, and learning theory and practice. In Part I, we look at problems in control of complex networks. In Chapter 1, we consider the performance analysis of a class of linear noisy dynamical systems. In Chapter 2, we look at the optimal design problems for these networks. In Chapter 3, we consider dynamical networks where interactions between the networks occur randomly in time. And in the last chapter of this part, in Chapter 4, we look at dynamical networks wherein coupling between the subsystems (or agents) changes nonlinearly based on the difference between the state of the subsystems. In Part II, we consider estimation problems wherein we deal with a large body of variables (i.e., at large scale). This part starts with Chapter 5, in which we consider the problem of sampling from a dynamical network in space and time for initial state recovery. In Chapter 6, we consider a similar problem with the difference that the observations instead of point samples become continuous observations that happen in Lebesgue measurable observations. In Chapter 7, we consider an estimation problem in which the location of a robot during the navigation is estimated using the information of a large number of surrounding features and we would like to select the most informative features using an efficient algorithm. In Part III, we look at active perception problems, which are approached using reinforcement learning techniques. This part starts with Chapter 8, in which we tackle the problem of multi-agent reinforcement learning where the agents communicate and classify as a team. In Chapter 9, we consider a single agent version of the same problem, wherein a layered architecture replaces the architectures of the previous chapter. Then, we use reinforcement learning to design the meta-layer (to select goals), action-layer (to select local actions), and perception-layer (to conduct classification).

Part I:

Control Problems in Complex Networked Systems

Chapter 1

Performance Analysis of Noisy Dynamical Networks

1.1 Introduction

Developing tools to reduce design complexity has been in the center of recent research in networked control systems [1–5]. In several important applications, network design problem reduces to finding an optimal communication (graph) topology among a network of identical subsystems that are coupled to each other through some common mission-related control objectives. Examples include formation control in a cooperative team of robots [6], the platoon of vehicles in automated highways [7], space-time rendezvous in a team of robots, and networks of synchronous oscillators in power networks. In fact, the recent jumps in the underlying technology have ignited several applications of these ideas [8, 9].

Here we bring a brief history of the origins of research in this field. From a control theoretic perspective, Tsitsiklis pioneered in systematically looking at the machinery required to tackle the decentralized coordination of the agents [10]. The relationship between this notion and the concept of *consensus* became more clear later. The seminal paper by Fax and Murray [11] is one of the earliest endeavors in the modern control that highlighted the connection of consensus problem to the Laplacian matrix of the underlying graph. Two significant extensions were carried out by Jadbabaie et. al. [12] and Olfati-Saber and Mur-

ray [13, 14], where the simple connectivity assumptions for the graph of the network were relaxed. These pieces have been either a foundation or inspiration for contributions in different directions. For instance, the consensus over random networks [15, 16] or time-varying topologies [17], rendezvous using proximity graphs [18], consensus of double-integrators [19], or with self-triggered communications [20] are a fraction of ideas that were followed by the community.

The design problem usually involves the optimization of a measure of performance or robustness while respecting various constraints. Due to their combinatorial nature, most network design problems become intractable as network size increases and suffer from high computational complexities. Possibility of characterizing performance and robustness measures in closed and explicit forms will significantly facilitate the design process by allowing the network designer to identify relevant functional properties of the measures and their behaviors with respect to the interconnection topology. In this chapter, we present explicit expressions for the $\mathcal{H}_2$-norm, as a performance and robustness measure, of a class of interconnected network of linear control systems.

The authors of [21] consider coherency of a platoon of vehicles by evaluating the $\mathcal{H}_2$-norm of second-order consensus algorithms and propose several scaling laws for various scenarios of coordination. In [22], ill-posedness of a certain class of platoons is investigated and shown that stabilizability deteriorates as the size of the platoon increases. The string stability of a class of formation problems with limited communication range is studied in [23], where a fundamental limit on the disturbance rejection quality of the network in the frequency domain is derived. The stability and robustness of large platoon of vehicles with double-integrator dynamics are considered in [24], where it is shown that how scaling of a robustness measure (in terms of the platoon size) to external disturbances improves from geometric to polynomial growth when vehicles are allowed to communicate with their two immediate neighbors. In [25], robustness analysis and distributed $\mathcal{H}_\infty$ controller design of platoon of vehicles with third-order models and undirected communication topologies are considered. In [26], several graph theoretic bounds on the $\mathcal{H}_2$-based performance of linear consensus networks with first- and second-order dynamics are characterized and it is shown how the performance measure scales with the network size and depends on structural

properties of the communication topology. In [27], the authors consider distributed $\mathcal{H}_2$ and $\mathcal{H}_\infty$ controller design for a multi-agent system whose subsystems have general linear time-invariant dynamics. Using a consensus-like algorithm and notion of the grounded graph (e.g., see [28]) to model coupling of agents to leaders, it is shown under what conditions such controllers exist and how they can be suboptimally designed.

In this Chapter, we consider a network of identical subsystems that are connected over an undirected graph and subject to external disturbance and measurement noise. We propose a methodology to express the steady-state variance of the output of a class of interconnected linear time-invariant networks as a rational function of their Laplacian eigenvalues. Our method extends the existing results in the literature for first- and second-order linear consensus network models (cf. [2] and reference in there). We illustrate that the notion of minimum connectivity threshold is useful for the design of the feedback gains for these networks. It turns out that stabilizability of the nodal dynamics (and detectability in case of observer-based output-feedback) guarantee the existence of such designs. Using these developments, it is shown that fundamental limits may emerge for networks whose subsystems are non-minimum phase. We find graph-theoretic bounds for the performance of the network, which paves the way to find scaling laws for the performance measure. Moreover, a tradeoff between the graph sparsity and performance measure is revealed. Additionally, for networks over path or cycle graphs, we find the asymptotic trend of the performance measure. We bring two extensions of the analyses for the cases of observer-based output feedback as well as a class of composite networks. We have included several parametric and numerical examples to support our theoretical contributions. Our approach is advantageous for the design of these dynamical networks. Our spectral expressions can facilitate solving of underlying optimal control problems: instead of dealing directly with optimization problems with high-dimensional matrices, our method leverages the structure of the control system and decouples the roles of typically low-dimensional feedback gains and the eigenspectrum of the communication graph. The proofs and additional details of the examples of this chapter are included in the appendix.

1.2 Notations and Preliminaries

Here we bring the notations and preliminaries used throughout this chapter. Most of these notations are adapted in the upcoming chapters as well.

The sets of real and integer numbers are denoted by $\mathbb{R}$ and $\mathbb{Z}$, respectively. The $_+$ and $_{++}$ subscripts denote the nonnegative and positive subsets of a set, respectively (e.g. $\mathbb{R}_+$). Tr(.) represents matrix trace. The partial ordering on the cone of positive-semidefinite matrices is denoted via $\succ$ and similar operators. The standard basis for $\mathbb{R}^N$ is denoted by $\{e_1, \ldots, e_N\}$. The vector and matrix of ones are denoted by $1_N \in \mathbb{R}^N$ and $\mathbf{J}_N \in \mathbb{R}^{N\times N}$, respectively. Also, $\mathbf{I}_N$ and $\mathbf{M}_N = \mathbf{I}_N - \mathbf{J}_N/N$ are identity and centering matrices, respectively. The centering matrix subtracts the average of a vector from every element; i.e., for any vector $x \in \mathbb{R}^N$, it holds that

$$\mathbf{M}_N\, x \;=\; x - \left(\frac{1}{N}\sum_{i=1}^{N} x_i\right) 1_N. \tag{1.1}$$

The vectorization is denoted by vec($\mathbf{S}$). The Kronecker product is denoted by $\mathbf{A} \otimes \mathbf{B}$. The matrix transpose and conjugate transpose are denoted by $(.)^T$ and $(.)^*$ superscripts, respectively. A weighted undirected graph over N nodes is a collection $\mathcal{G} = (\mathcal{V}, \mathcal{E}, k)$, with a set of nodes $\mathcal{V} = \{1, 2, \ldots, N\}$, a set of edges $\mathcal{E} \subset \big\{\{i, j\} : i, j \in \mathcal{V}\big\}$, and a weight function $k :\ \mathcal{E} \to \mathbb{R}_+$. We define $k_{ij} := k(\{i, j\}) = k_{ji}$ and form the (symmetric) graph Laplacian $\mathbf{L} = [l_{ij}] \in \mathbb{R}^{N\times N}$ with entries

$$l_{ij} = \begin{cases} \displaystyle\sum_{\{i,j\}\in\mathcal{E}} k_{ij} & \text{if } i = j \\ -\,k_{ij} & \text{if } i \neq j \end{cases}. \tag{1.2}$$

The set of neighbors of a node is $\mathcal{N}_i := \big\{j \in \mathcal{V} \;\big|\; \{i, j\} \in \mathcal{E}\big\}$ for $i \in \mathcal{V}$. The eigenvalues of $\mathbf{L}$ are denoted by $\lambda_1 \leq \cdots \leq \lambda_N$, which are real and nonnegative for a weighted undirected graph. For a connected graph, $\lambda_1 = 0$ with eigenvector 1_N, and $\lambda_2 > 0$. The Laplacian eigendecomposition is $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is its orthonormal matrix of eigenvectors and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$.

Consider two positive sequences $\{p_n\}_{n\in\mathbb{Z}_+}$ and $\{q_n\}_{n\in\mathbb{Z}_+}$. To compare their asymptotic

behavior, consider the following notations.

$$q_n = O(p_n) \text{ if } q_n/p_n \leq C \text{ for some } C > 0.$$

$$q_n = O(p_n) \Leftrightarrow p_n = \Omega(q_n).$$

$$q_n = O(p_n),\ q_n = \Omega(p_n) \Leftrightarrow q_n = \Theta(p_n).$$

$$q_n \sim p_n \Leftrightarrow \lim_{n\to\infty} q_n/p_n = 1.$$

1.3 Problem Statement

We consider an interconnected network of N subsystems where the dynamics of the i'th subsystem are governed by

$$S_i : \begin{cases} \dot{x}_i(t) &= \mathbf{A}\, x_i(t) + \mathbf{B}\, u_i(t) + \mathbf{E}\, \xi_i(t) \\ y_i(t) &= \mathbf{H}\, x_i(t) + \sigma\, \eta_i(t) \\ z_i(t) &= \mathbf{C}\, x_i(t), \end{cases} , \tag{1.3}$$

for $i = 1, \ldots, N$, in which $x_i(t) \in \mathbb{R}^n$ is the state vector of the subsystem, $u_i(t) \in \mathbb{R}^p$ is the control input, $\xi_i(t) \in \mathbb{R}^{m_1}$ is an exogenous disturbance input, $\eta_i(t) \in \mathbb{R}^{m_3}$ is the measurement noise, $y_i(t) \in \mathbb{R}^q$ is the measurable output, and $z_i(t) \in \mathbb{R}^{m_2}$ is the performance output. The matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{B} \in \mathbb{R}^{n\times p}$, $\mathbf{C} \in \mathbb{R}^{m_2\times n}$, $\mathbf{E} \in \mathbb{R}^{n\times m_1}$, and $\mathbf{H} \in \mathbb{R}^{q\times n}$ are fixed known matrices. Parameter $\sigma \geq 0$ dictates the magnitude of the measurement noise compared to the disturbance process. The state of the entire network is

$$x(t) := \left[x_1(t)^T, x_2(t)^T, \ldots, x_N(t)^T\right]^T \in \mathbb{R}^{Nn}.$$

The vectors representing the network input, disturbance, feedback noise, feedback output, and controlled output are similarly defined and denoted by u, ξ, η, y, and z, respectively.

The control objective for the network is to achieve asymptotic synchronization (or con-

sensus), i.e., to fulfil the goal

$$x_i(t) - x_j(t) \to 0 \text{ as } t \to \infty$$

for all $i, j \in \{1, \ldots, N\}$. To realize this objective, we employ the following feedback control law

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} \mathbf{K}_{ij} \left(y_i(t) - y_j(t) \right) \tag{1.4}$$

for each subsystem $i \in \{1, \ldots, N\}$. The subsystems are allowed to exchange their relative output measurements information over an undirected communication graph $\mathcal{G}$. It is assumed that the structure of the feedback gain matrices $\mathbf{K}_{ij} \in \mathbb{R}^{p \times q}$ are restricted to

$$\mathbf{K}_{ij} = k_{ij} \mathbf{K}$$

where k_{ij}'s are nonnegative scalars (i.e., the weights of graph $\mathcal{G}$) and $\mathbf{K} \in \mathbb{R}^{ptimesq}$ is the common factor among all feedback gain matrices.

When stabilizing feedback control law (1.4) exists and there is no exogenous noise, i.e., $\mathbf{E} = \mathbf{0}$ and $\sigma = 0$, one can show that $x_i(t) - x_j(t) \to 0$ as $t \to \infty$ holds for the closed-loop network. However, in the presence of noise, the state variables will fluctuate around the consensus state. Let us quantify these fluctuations. First, note that the deviation from the average of state of a subsystems is

$$\nu_i(t) \ := \ z_i(t) - \frac{1}{N} \sum_{j=1}^{N} z_i(t) \tag{1.5}$$

for every node $i \in \mathcal{V}$. We can represent (1.5) in vector form as

$$\nu(t) = \left(\mathbf{M}_N \otimes \mathbf{I}_{m_1} \right) z(t) = \left(\mathbf{M}_N \otimes \mathbf{C} \right) x(t), \tag{1.6}$$

where $\mathbf{M}_N$ is the centering matrix of size N. We inspect that the network (1.3) with control law (1.4) asymptotically reaches consensus *if and only if* vector of deviations from

the average $\nu(t)$ asymptotically goes to zero. Since control law (1.4) can be rewritten as

$$u(t) = -(\mathbf{L} \otimes \mathbf{KH})\, x(t), \tag{1.7}$$

the controller synthesis breaks into two components: designing a feedback gain $\mathbf{K}$ and designing a weighted undirected graph with Laplacian $\mathbf{L}$. It is assumed that measurement noise and disturbance are both Gaussian, uncorrelated, and with independent components with unit variance.In order to measure the aggregate fluctuations in the network, we adopt the steady-state variance of the deviation from the average as a measure of performance for the design, which is defined by

$$\rho(\mathbf{L}, \mathbf{K}) := \lim_{t \to \infty} \mathbb{E}\left\{ \|\nu(t)\|_2^2 \right\}. \tag{1.8}$$

The *research problems* are to characterize performance measure (1.8) in terms of Laplacian eigenvalues of the underlying communication graph of the network, illustrate role of feedback (and observer) gains in stability and emergence of fundamental limits on performance measure and design tradeoffs, and derive scaling laws for the performance as the network grows.

1.4 Stability and Performance Measure Characterization

We look at the stability criteria for these dynamical networks. Moreover, we derive and characterize spectral expressions for the performance measure. For brevity, we remove the time argument from the variables in the rest of this chapter.

Once we apply feedback control protocol (1.4), the closed-loop dynamics of the state of the network are given by

$$\dot{x} = (\mathbf{I}_N \otimes \mathbf{A} - \mathbf{L} \otimes \mathbf{BKH})\, x + (\mathbf{I}_n \otimes \mathbf{E})\, \xi - (\mathbf{L} \otimes \sigma \mathbf{I}_{m_3})\eta. \tag{1.9}$$

We define auxiliary variables r, χ, and γ to be

$$r := (\mathbf{U}^T \otimes \mathbf{I}_n)\, x,\ \chi := (\mathbf{U}^T \otimes \mathbf{I}_{m_1})\, \xi, x,\ \gamma := (\mathbf{U}^T \otimes \mathbf{I}_{m_3})\eta. \tag{1.10}$$

Then, the following dynamical decoupling is realized (see [27] for the case of state-feedback without the measurement noise).

Proposition 1.4.1. *By the change of variables (1.10), the resulting closed-loop network dynamics given by (1.9) are decoupled into N systems*

$$\Sigma_i : \qquad \dot{r}_i \ = \ (\mathbf{A} - \lambda_i\, \mathbf{BKH})\, r_i + \mathbf{E}\, \chi_i - \lambda_i \mathbf{BK}\sigma\gamma_i, \tag{1.11}$$

for each $i = 1, 2, \ldots, N$. In the absence of disturbance and noise, the network reaches consensus if and only if systems $\Sigma_2, \ldots, \Sigma_N$ are asymptotically stable.

We leverage this decoupling to arrive at spectral expressions for the performance measure of the network.

Theorem 1.4.2. *Suppose that in* (1.11) *systems $\Sigma_2, \ldots, \Sigma_N$ are asymptotically stable. Then, the performance measure can be expressed as*

$$\rho(\mathbf{L}, \mathbf{K}) = \sum_{i=2}^{N} \phi(\lambda_i, \mathbf{K}), \tag{1.12}$$

with the performance function $\phi(\lambda, \mathbf{K})$ given by

$$\phi(\lambda, \mathbf{K}) := \mathrm{Tr}\left(\mathbf{CP}(\lambda, \mathbf{K})\mathbf{C}^T\right), \tag{1.13}$$

which is a rational function of λ and entries of $\mathbf{K}$. The map $\mathbf{P}(\lambda, \mathbf{K})$ is the unique positive-definite solution to an algebraic Lyapunov equation given by

$$(\mathbf{A} - \lambda\mathbf{BKH})\mathbf{P}(\lambda, \mathbf{K}) + \mathbf{P}(\lambda, \mathbf{K})(\mathbf{A} - \lambda\mathbf{BKH})^T + \mathbf{EE}^T + \lambda^2\sigma^2\mathbf{BK}(\mathbf{BK})^T = \mathbf{0}, \tag{1.14}$$

for all values of λ that make $\mathbf{A} - \lambda\mathbf{BKH}$ a Hurwitz matrix.

The dimension of the dynamics of each subsystem is often small and has nothing to do with the number of subsystems. Therefore, evaluation of performance function $\phi(\lambda, \mathbf{K})$ can be done via symbolically solving Lyapunov equation (1.14) after converting it to a linear system by vectorization (see the proof of Theorem 1.4.2).

Due to linearity of the Lyapunov equation in terms $\mathbf{E}\mathbf{E}^T + \lambda^2\sigma^2\mathbf{B}\mathbf{K}(\mathbf{B}\mathbf{K})^T$ one inspects that the performance function ϕ can be decomposed into two components according to

$$\phi(\lambda, \mathbf{K}) = \phi_\xi(\lambda, \mathbf{K}) + \sigma^2\phi_\eta(\lambda, \mathbf{K}),$$

in which spectral functions ϕ_ξ and ϕ_η only reflect the effect of disturbance and measurement noise, respectively.

Remark 1. In this Chapter, we occasionally skip argument $\mathbf{K}$ in $\phi(\lambda, \mathbf{K})$ and denote it as $\phi(\lambda)$. In those cases, we solely consider the dependence of the functions on the eigenvalues of the graph Laplacian (i.e., for a fixed feedback gain $\mathbf{K}$).

Remark 2. A part of the result of Theorem 1.4.2 is hidden in the analysis provided in [27], in the case of state-feedback. However, the authors did not explicitly derive the spectral expressions for the performance.

1.4.1 Extension of Stability Analysis to Observer Design

We extend the previous analysis to the output-feedback and synthesize a decentralized observer. We show that the separation principal in the linear filtering using Luenberger observers is naturally carried into this design as well.

Our procedure consists of four steps:

(i) We augment the dynamics of subsystem i by an observer variable $\hat{x}_i \in \mathbb{R}^n$, whose dynamics are governed by

$$\dot{\hat{x}}_i = \mathbf{A}\hat{x}_i + \mathbf{B}u_i + \hat{u}_i, \tag{1.15}$$

where $\hat{u}_i \in \mathbb{R}^n$ is an auxiliary control input for the observer. We will set the value of this input in a decentralized manner in the last step.

(ii) As it is usual in the observer design, we use $\hat{x}_i$ to compute

$$u_i = -\mathbf{K}\hat{x}_i. \tag{1.16}$$

(iii) In addition to the relative output feedback on $\mathbf{H}x_i$, the subsystems should share the value of $\mathbf{H}\hat{x}_i$ with their neighbors. Once we consider these three steps, the augmented dynamics of subsystem i are given by

$$\hat{S}_i : \begin{cases} \begin{bmatrix} \dot{x}_i \\ \dot{\hat{x}}_i \end{bmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{BK} \\ \mathbf{0} & \mathbf{A}-\mathbf{BK} \end{bmatrix} \begin{bmatrix} x_i \\ \hat{x}_i \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_n \end{bmatrix} \hat{u}_i + \begin{bmatrix} \mathbf{E} \\ \mathbf{0} \end{bmatrix} \xi_i \\ \hat{y}_i = \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{bmatrix} x_i \\ \hat{x}_i \end{bmatrix} + \begin{bmatrix} \mathbf{I}_{m_3} \\ \mathbf{0} \end{bmatrix} \eta_i \\ z_i = \begin{bmatrix} \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_i \\ \hat{x}_i \end{bmatrix} \end{cases}, \tag{1.17}$$

Variable $\hat{y}_i$ has the same role as y_i in (1.3); i.e., the augmented subsystems will use the relative-feedback on this variable.

(iv) We use the following theorem and design the gain of control law (1.4) when applied on subsystems $\hat{S}_i$ in (1.17), which in this case will be an observer gain.

Theorem 1.4.3. *Suppose that we apply control law (1.4) on augmented subsystems $\hat{S}_i$ in (1.17) by setting*

$$\hat{u}_i = -\hat{\mathbf{F}} \sum_{j \in \mathcal{N}_i} a_{ij}(\hat{y}_i - \hat{y}_j), \tag{1.18}$$

where the observer gain is set to be

$$\hat{\mathbf{F}} = \begin{bmatrix} -\mathbf{F}, & \mathbf{F} \end{bmatrix} \in \mathbb{R}^{n \times (2q)}. \tag{1.19}$$

Moreover, assume that $\mathbf{F} \in \mathbb{R}^{n \times q}$ is chosen such that $\mathbf{A} - \lambda_i \mathbf{FH}$ is Hurwitz for $i = 2, \ldots, N$. Then, the estimation and regulation are separated: if we apply control input u_i given in (1.16) *for any $\mathbf{K}$ that makes $\mathbf{A}-\mathbf{BK}$ a Hurwitz matrix, then the network with this observer-*

based relative output-feedback reaches the consensus in the absence of disturbance and noise.

For this design, we denote the resulting performance function by $\phi(\lambda, \mathbf{K}, \mathbf{F})$. This function can be found similar to the case of simple state-feedback, except that we need the augmented matrices of $\hat{S}_i$ given in (1.17) for solving (1.14) and evaluation of this function.

Remark 3. Note that the separation principal only holds in the stability analysis. The resulting performance functions $\phi(\lambda, \mathbf{K}, \mathbf{F})$ may not be decomposed into functions that only depend on one of gains $\mathbf{K}$ or $\mathbf{F}$.

The separation principal together with the duality between the estimation and regulation let us prove similar results for the quality of estimation using this decentralized observer. We now elaborate. First, we define the error of estimation using this observer as

$$e(t) \;:=\; \hat{x}(t) - x(t). \tag{1.20}$$

Because we are only employing the relative feedback, we may only control the deviations of the error components from their average. These deviations are reflected by the variable

$$\delta(t) := (\mathbf{M}_N \otimes \mathbf{I}_n)\, e(t). \tag{1.21}$$

Next, we define the estimation measure for network as

$$\mu(\mathbf{L}, \mathbf{F}) := \lim_{t\to\infty} \mathbb{E}\left\{\|\delta(t)\|_2^2\right\}. \tag{1.22}$$

The dual of system Σ_i in (1.11) is

$$\Upsilon_i : \qquad \dot{r}_i \;=\; (\mathbf{A} - \lambda_i\, \mathbf{F}\mathbf{H})\, r_i + \mathbf{E}\, \chi_i - \lambda_i \sigma \mathbf{F} \gamma_i \quad , \tag{1.23}$$

which lets us deduce the next result (compare to Theorem 1.4.2).

Theorem 1.4.4. *Suppose that in* (1.23) *systems* $\Upsilon_2, \ldots, \Upsilon_N$ *are asymptotically stable.*

Then, we can express the estimation measure as

$$\mu(\mathbf{L}, \mathbf{F}) = \sum_{i=2}^{N} \psi(\lambda_i, \mathbf{K}), \tag{1.24}$$

with the estimation function $\psi(\lambda, \mathbf{K})$ *given by*

$$\psi(\lambda, \mathbf{K}) := \mathrm{Tr}\left(\mathbf{Q}(\lambda, \mathbf{K})\right), \tag{1.25}$$

which is a rational function of λ *and entries of* $\mathbf{F}$*. The map* $\mathbf{Q}(\lambda, \mathbf{K})$ *is the unique positive-definite solution to an algebraic Lyapunov equation given by*

$$(\mathbf{A} - \lambda\mathbf{F}\mathbf{H})\mathbf{Q}(\lambda, \mathbf{K}) + \mathbf{Q}(\lambda, \mathbf{K})(\mathbf{A} - \lambda\mathbf{F}\mathbf{H})^T + \mathbf{E}\mathbf{E}^T + \lambda^2\sigma^2\mathbf{F}^T\mathbf{F} = 0, \tag{1.26}$$

for all values of λ *that make* $\mathbf{A} - \lambda\mathbf{F}\mathbf{H}$ *a Hurwitz matrix.*

Remark 4. In [29], the authors propose the following alternative observer-based approach. They define their observer variable v_i to follow the dynamics

$$\dot{v}_i = \mathbf{F}v_i + \mathbf{G}y_i + \mathbf{T}\mathbf{B}u_i, \tag{1.27}$$

and define their control law as

$$u_i = \mathbf{K}\mathbf{Q}_1 \sum_{i=1}^{N} a_{ij}(y_i - y_j) + \mathbf{K}\mathbf{Q}_2 \sum_{i=1}^{N} a_{ij}(v_i - v_j), \tag{1.28}$$

where matrix $\mathbf{F}$ has no eigenvalue in common with $\mathbf{A}$, the pair $(\mathbf{F}, \mathbf{G})$ is stabilizable, and $\mathbf{T}$ is the unique solution to Sylvester equation $\mathbf{T}\mathbf{A} - \mathbf{F}\mathbf{T} = \mathbf{G}\mathbf{C}$. Then, they design $\mathbf{K}, \mathbf{Q}_1$ and $\mathbf{Q}_2$ such that a design with minimum connectivity threshold is achieved. One can see that our design is different and simpler as we only need a feedback gain $\mathbf{K}$ and an observer gain $\mathbf{F}$. Moreover, our approach is built upon the separation principle between the regulation and estimation, which is also the case in the classical Luenberger (or LQG) observer design. Therefore, unlike our approach, It is not evident how the separation principal shows up in their design.

1.5 Design of Control Law Gains

We investigate the problem of finding feedback gains and focus on gains inducing a minimum connectivity threshold. This property makes the design process with respect to the graph more tractable. After that, we discuss related performance limitations.

1.5.1 Minimum Connectivity Threshold

We define the minimum connectivity threshold $\tilde{\lambda}(\mathbf{K}) \in [0, \infty]$ for a feedback gain $\mathbf{K}$ to be

$$\tilde{\lambda}(\mathbf{K}) := \inf_{\lambda>0} \big\{\lambda : (\mathbf{A} - c\,\mathbf{BKH}) \text{ is Hurwitz for } c > \lambda\big\}. \tag{1.29}$$

Similar notions have been reported (e.g. [27]), while our goal is characterization of conditions for finding gains with $\tilde{\lambda}(\mathbf{K}) < \infty$[1]. The following definition is for this purpose.

Definition 1.5.1. *The feedback gain* $\mathbf{K}$ *is said to have an unbounded stability region if* $\tilde{\lambda}(\mathbf{K}) \in [0, \infty)$.

If $\mathbf{K}$ has an unbounded stability region, then the network is robust to all increases in the connectivity: if the network is output-stable for a given graph $\mathcal{G}_1$ with Laplacian $\mathbf{L}_1$, then for every graph $\mathcal{G}_2$ with Laplacian $\mathbf{L}_2$ and $\mathcal{G}_1 \subset \mathcal{G}_2$, the network is still output-stable. The reason is that $\lambda_i(\mathbf{L}_1) \leq \lambda_i(\mathbf{L}_2)$ for $i = 2, \ldots, N$ (this has been emphasized in [27] as well). Moreover, this makes the stability analysis with respect to the graph more tractable, since ensuring $\lambda_2(\mathbf{L}) > \tilde{\lambda}(\mathbf{K})$ guarantees the output-stability of network. Before bringing methods to find such feedback gains, let us look at a consequence of choosing them.

Theorem 1.5.2. *For a network designed with a feedback gain* $\mathbf{K}$ *that in endowed by a connectivity threshold* $\tilde{\lambda}(\mathbf{K}) < \infty$*, the performance function* $\phi(\lambda)$ *is analytic on interval* $(\tilde{\lambda}(\mathbf{K}), \infty)$.

The openness of the interval of interest in Theorem 1.5.2 suggests that if $\tilde{\lambda}(\mathbf{K}) > 0$, we need to maintain a minimum distance from this value. This will make sure that the stability margin is large enough.

[1] $\tilde{\lambda}(\mathbf{K}) = \infty$ corresponds to finding the infimum of the empty set in (1.29).

1.5.2 State-Feedback Minimum Connectivity Design

Let us consider the state-feedback (i.e., $\mathbf{H} = \mathbf{I}_n$ in (1.3)). It turns out that the stabilizability is the necessary and sufficient condition for existence a gain $\mathbf{K}$ that induces a bounded threshold $\tilde{\lambda}(\mathbf{K})$.

Theorem 1.5.3. *If $(\mathbf{A}, \mathbf{B})$ is stabilizable, then for every value of $c > 0$, the choice of feedback gain given by*

$$\mathbf{K} = \frac{1}{2}\mathbf{B}^T\mathbf{Q}^{-1}, \tag{1.30}$$

satisfies $\tilde{\lambda}(\mathbf{K}) \in [0, c]$, where $\mathbf{Q} \succ 0$ is a solution to the following feasible linear matrix inequality.

$$\mathbf{AQ} + \mathbf{QA}^T - c\,\mathbf{BB}^T \prec \mathbf{0}. \tag{1.31}$$

Conversely, if there exists a gain $\mathbf{K}$ with $\tilde{\lambda}(\mathbf{K}) < \infty$, then $(\mathbf{A}, \mathbf{B})$ is stabilizable.

The linear matrix inequality (LMI) (1.31) is a computational tool to find a gain $\mathbf{K}$ for a given network and graph with a minimum connectivity threshold at most equal to c (see Example 1.9.1). The solvability of LMI (1.31) is called the quadratic stabilizability of $(\mathbf{A}, \mathbf{B})$ by means of a linear state-feedback (see Section 7.2 of [30]).

Remark 5. This result is inspired by Theorem 11 in [27], while our main contribution is to clarify the role of stabilizability in existence of feedback gains with minimum connectivity threshold.

Remark 6. The optimality in choice of $\mathbf{Q}$ is not the concern in Theorem 1.5.3. Instead, we focus on the *existence* of designs for $\mathbf{K}$ with a minimum connectivity design. In fact, various performance criteria could potentially get addressed. For instance, suppose that for some $d > 0$, we replace LMI (1.31) with

$$\mathbf{AQ} + \mathbf{QA}^T - c\mathbf{BB}^T + 2d\mathbf{Q} \prec \mathbf{0}. \tag{1.32}$$

Then, for $\mathbf{K}$ computed from (1.30) using any solution to this inequality $\mathbf{Q} \succ \mathbf{0}$, not only

$\tilde{\lambda}(\mathbf{K}) \leq c$, but also for each eigenvalue $\lambda > \tilde{\lambda}(\mathbf{K})$, the poles of $\mathbf{A} - \lambda\mathbf{BK}$ have real parts less than $-d$ (see [31]). As another example, authors of [27] brought a version of the matrix inequality which ensures that each decoupled subsystem Σ_i has $\mathcal{H}_2$-norm less than a desired value, which they state that could be conservative in practice. Criteria such as robustness or non-fragility could be potentially added by building on top of (1.31) as well (e.g. see [32]).

1.5.3 Observer-Based Minimum Connectivity Design for Output-Feedback

The duality between the derived conditions on $\mathbf{A}-\lambda_i\mathbf{FH}$ in Theorem 1.4.3 and on $\mathbf{A}-\lambda_i\mathbf{BK}$ in Theorem 1.5.3 lets us conclude the following result that resembles the result of Theorem 1.5.3.

Theorem 1.5.4. *Suppose that* $(\mathbf{A}, \mathbf{H})$ *is detectable. Then, for every* $c > 0$*, the following observer gain for the settings of Theorem 1.4.3, has an unbounded stability region with* $\tilde{\lambda}(\mathbf{F}) \in [0, c]$.

$$\hat{\mathbf{F}} = \left[-\frac{1}{2}\mathbf{Q}^{-1}\mathbf{H}^T,\ \frac{1}{2}\mathbf{Q}^{-1}\mathbf{H}^T\right] \in \mathbb{R}^{n\times(2q)}, \tag{1.33}$$

where $\mathbf{Q} \succ 0$ *is a solution to the following feasible LMI.*

$$\mathbf{A}^T\mathbf{Q} + \mathbf{QA} - c\,\mathbf{H}^T\mathbf{H} \prec \mathbf{0}. \tag{1.34}$$

Conversely, if under the settings of Theorem 4 an observer gain $\mathbf{F}$ *has a bounded* $\tilde{\lambda}(\mathbf{F})$*, then* $(\mathbf{A}, \mathbf{H})$ *is detectable.*

The LMI (1.34) is the quadratic stabilizability condition for the dual pair $(\mathbf{A}^T, \mathbf{H}^T)$ (it is stabilizable since $(\mathbf{A}, \mathbf{H})$ is detectable).

1.5.4 Asymptotic Performance and Estimation Bounds

An important design question is if the performance function $\phi(\lambda, \mathbf{K})$ can be made arbitrarily small, which is related to the notion of almost disturbance decoupling [33]: attenuating the effect of the disturbance in a performance metric as much as desired. We study the case of relative state-feedback below.

Theorem 1.5.5. *Suppose that* $(\mathbf{A}, \mathbf{B})$ *is stabilizable and* $(\mathbf{A}, \mathbf{C})$ *is detectable and that* $\sigma = 0$*. For all pairs of* $\lambda > 0$ *and* $\mathbf{K}$ *for which* $\mathbf{A} - \lambda\mathbf{B}\mathbf{K}$ *is Hurwitz, the performance function resulting from the relative state-feedback is bounded from below according to*

$$\phi(\lambda, \mathbf{K}) > \mathrm{Tr}\Big(\mathbf{E}^T\mathbf{P}_0\mathbf{E}\Big), \tag{1.35}$$

for a positive semi-definite matrix $\mathbf{P}_0$ *given by*

$$\mathbf{P}_0 := \lim_{\epsilon \to 0} \mathbf{P}_\epsilon \tag{1.36}$$

where $\mathbf{P}_\epsilon$ *is the unique positive semi-definite solution to the parametric algebraic Riccati equation*

$$\mathbf{A}^T\mathbf{P}_\epsilon + \mathbf{P}_\epsilon\mathbf{A} + \mathbf{C}^T\mathbf{C} - \epsilon^{-2}\mathbf{P}_\epsilon\mathbf{B}\mathbf{B}^T\mathbf{P}_\epsilon = \mathbf{0}. \tag{1.37}$$

Matrix $\mathbf{P}_0$ *is zero if and only if transfer matrix* $\mathbf{C}(s\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}$ *is right-invertible and minimum-phase.*

For instance, if the transfer matrix is $\mathbf{C}(s\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}$ non minimum-phase and the columns of $\mathbf{E}$ are not in the null space of $\mathbf{P}_0$, then the bound in (1.35) is strictly positive. The dual of this result for estimation quality is given below, whose proof is identical to Theorem 1.5.5 and has been omitted.

Theorem 1.5.6. *Suppose that* $(\mathbf{A}, \mathbf{E})$ *is stabilizable and* $(\mathbf{A}, \mathbf{H})$ *is detectable. If for some gain* $\mathbf{F}$*,* $\mathbf{A} - \lambda_i\mathbf{F}\mathbf{H}$ *are Hurwitz for* $i = 2, \ldots, N$*, then*

$$\psi(\lambda, \mathbf{F}) > \mathrm{Tr}(\mathbf{S}_0),$$

for a positive semi-definite matrix $\mathbf{S}_0$ *given by*

$$\mathbf{S}_0 := \lim_{\sigma \to 0} \mathbf{S}_\sigma, \tag{1.38}$$

where $\mathbf{S}_\sigma$ *is the unique positive semi-definite solution to the parametric algebraic Riccati*

equation

$$\mathbf{A}\mathbf{S}_\sigma + \mathbf{S}_\sigma\mathbf{A}^T + \mathbf{E}\mathbf{E}^T - \sigma^{-2}\mathbf{S}_\sigma\mathbf{H}^T\mathbf{H}\mathbf{S}_\sigma = \mathbf{0}; \tag{1.39}$$

Matrix $\mathbf{S}_0$ *is zero if and only if transfer matrix* $\mathbf{H}(s\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{E}$ *is right-invertible and minimum-phase.*

1.5.5 Parametric Evaluation of $\tilde{\lambda}(\mathbf{K})$

In both relative state or output feedback designs, if n is not large (e.g. $n \sim 1$ to 4), we may design $\mathbf{K}$ with an unbounded stability region using Routh-Hurwitz criteria and explicitly evaluate $\tilde{\lambda}(\mathbf{K})$. In fact, the characteristic equation of the matrix $\mathbf{A} - \lambda\mathbf{B}\mathbf{K}\mathbf{H}$ for the decoupled systems for eigenvalue λ is

$$p_\lambda(s) = p(s; \lambda, \mathbf{K}) = \det\left(s\,\mathbf{I}_n - (\mathbf{A} - \lambda\mathbf{B}\mathbf{K}\mathbf{H})\right). \tag{1.40}$$

They must be Hurwitz polynomials for $\lambda = \lambda_2, \ldots, \lambda_N$. As we enforce the Routh-Hurwitz criteria, we find a set of essentially nonlinear inequalities involving λ and elements of $\mathbf{K}$, such that the minimum connectivity threshold is realizable and evaluable based on values of $\mathbf{K}$ (see the next section for examples).

1.6 Examples of Performance Analysis

In this section, before bringing more theoretical contributions, we bring different classes of subsystems and characterize their performance within this framework. Note that more details from these examples have been provided in Appendix Q. In many examples in this chapter, we are interested in convexity of the performance functions. This is due to the fact that this property would let us derive graph theoretic performance bounds for the network (see Section 1.8).

First, we consider two single-input single-output controllable subsystems under the relative state-feedback, where the disturbance and control input drive the dynamics from the same channel (without the measurement noise).

Realization	$\phi(\lambda, \mathbf{K})$
$\mathfrak{s}_1$: $\mathbf{A} = -a,$ $\mathbf{B} = \mathbf{E} = 1,\ \mathbf{C} = 1$	$\dfrac{1}{2(k\lambda + a)}$
$\mathfrak{s}_2$: $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix}$ $\mathbf{B} = \mathbf{E} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ $\mathbf{C} = [b_1\ b_0]$	$\dfrac{b_0^2 k_1 \lambda + a_2 b_0^2 + b_1^2}{2(k_2\lambda + a_1)(k_1\lambda + a_2)}$

Table 1.1: The subsystems investigated in Example 1.6.1 together with the performance functions in the case of relative state-feedback with $\sigma = 0$. We assume that $a, a_1, a_2 \geq 0$.

Example 1.6.1. Consider the subsystems given in Table 1.1, where we have also reported the corresponding performance functions. For the nodal dynamics $\mathfrak{s}_1$ and $\mathfrak{s}_2$, supposing that $\mathbf{K} = k > 0$ and $\mathbf{K} = [k_1, k_2] \succ 0$, respectively, in both cases $\tilde{\lambda}(\mathbf{K}) = 0$. Moreover, for $\lambda > \tilde{\lambda}(\mathbf{K})$, performance function $\phi(\lambda)$ is strictly convex and strictly decreasing. If all a_i's are zero and $\mathbf{C} = e_1^T$, these subsystems are called single and double-integrators, respectively. As a numerical example, let us consider double-integrators with $k_1 = k_2 = 1$. Then, using the second row of Table 1.1 we get

$$\phi(\lambda) = \frac{1}{2\lambda^2}. \tag{1.41}$$

This is a well-known result (e.g. see [34]).

Example 1.6.2. Consider double-integrators with relative feedback only on positions; i.e.,

$$\mathbf{H} = [1, 0].$$

We use the decentralized observer of Theorem 1.5.4. We let $\mathbf{K} = [k_1, k_2] \succ 0$ and set the observer gain to be $\mathbf{F} = [f_1, f_2]^T$. Theorem 1.4.3 requires the stability analysis for matrix $\mathbf{A} - \lambda \mathbf{F}\mathbf{H}$, which is Hurwitz if and only if $f_1, f_2 > 0$. Then, we get $\tilde{\lambda}(\mathbf{F}) = 0$. We can show that

$$\phi(\lambda, \mathbf{K}, \mathbf{F}) = \frac{c_1\lambda^4 + c_2\lambda^3 + c_3\lambda^2 + c_4\lambda + c_5}{c_6\lambda^4 + c_7\lambda^3 + c_8\lambda^2}, \tag{1.42}$$

where c_1 to c_8 are polynomials of k_1, k_2, f_1, and f_2. Using the observer with $k_1 = k_2 = f_1 = f_2 = 1$, (1.42) becomes

$$\phi(\lambda) = \frac{9\lambda^4 + 11\lambda^3 + 9\lambda^2 + 4\lambda + 1}{6\lambda^4 + 2\lambda^2}. \tag{1.43}$$

One observes that for weak connectivity regimes (i.e., λ near zero), $\phi(\lambda)$ in (1.43) is close to the function in (1.41), while as λ increases, the performance function corresponding to relative state-feedback vanishes, while the function from observer design does not.

Example 1.6.3. We consider a triple-integrator subsystem with dynamics

$$\dddot{x}_i = u_i + \xi_i. \tag{1.44}$$

Let us choose the state to be $[x_i, \dot{x}_i, \ddot{x}_i]^T$ with element-wise positive gain $\mathbf{K} = [k_1, k_2, k_3] \succ 0$. We can show that

$$\phi(\lambda, \mathbf{K}) = \frac{k_3}{2(k_1 k_2 k_3 \lambda^2 - k_1^2 \lambda)},\ \tilde{\lambda}(\mathbf{K}) = \frac{k_1}{k_2 k_3}. \tag{1.45}$$

The next two examples also have performance functions that under conditions become strictly decreasing and convex.

Example 1.6.4. The dynamics of a harmonic oscillator of mass m are governed by

$$\ddot{x}_i = -2\zeta\omega_0 \dot{x}_i - \omega_0^2 x_i + \frac{u_i}{m} + \frac{\xi_i}{m}, \tag{1.46}$$

where ζ is the damping ratio and ω_0 is the undamped angular frequency (see [35]). We consider $\mathbf{C} = [1, 0]$ and compute $\phi(\lambda)$ with the relative state-feedback on $[x, \dot{x}]^T$ with $\mathbf{K} = [k_1, k_2] \succ 0$. Using arguments similar to Example 1.6.1, if we define $\alpha_1 := m\omega_0^2/k_1$ and $\alpha_2 := 2m\omega_0\zeta/k_2$ we get the performance function

$$\phi(\lambda, \mathbf{K}) = \frac{1}{2k_1 k_2 \left(\lambda + \alpha_1\right)\left(\lambda + \alpha_2\right)}. \tag{1.47}$$

Again, for element-wise positive feedback gains, $\phi(\lambda)$ is strictly convex and strictly decreas-

ing for $\lambda > \tilde{\lambda}(\mathbf{K}) = 0$.

Example 1.6.5 (Platoon of Vehicles). We consider a network of vehicles, in which the position of i'th vehicle is denoted by $p_i \in \mathbb{R}$. It has the third-order dynamics

$$\tau \dddot{p}_i + \ddot{p}_i = u_i + \xi_i, \tag{1.48}$$

where the input $u_i \in \mathbb{R}$ is the desired acceleration and $\xi_i \in \mathbb{R}$ is the disturbance. The time-constant $\tau > 0$ characterizes how fast the vehicles responds to the acceleration command. The state vector is chosen as $[p_i, \dot{p}_i, \ddot{p}_i]^T$, where they denote the (errors in) the position, velocity, and acceleration of the vehicles in the platoon, respectively (see [25] for more details). The state-space matrices are given in the appendix. Using relative state-feedback, by application of the Routh-Hurwitz criteria we find that if $\mathbf{K} = [k_1, k_2, k_3]$ satisfies $k_1, k_2 > 0, k_3 \geq 0$, we get

$$\tilde{\lambda}(\mathbf{K}) = \begin{cases} 0 & \text{if } k_3 = 0 \\ \max\left\{0, \dfrac{\tau k_1 - k_2}{k_2 k_3}\right\} & \text{if } k_3 > 0 \end{cases}. \tag{1.49}$$

We can show that if $\sigma = 0$, we get

$$\phi(\lambda, \mathbf{K}) = \frac{1}{2k_1 k_2} \frac{k_3\lambda + 1}{k_3\lambda^3 + (k_2 - k_1\tau)\lambda^2/k_2}. \tag{1.50}$$

If $k_3 = 0$, the design corresponds to a relative output-feedback on only positions and velocities, with a performance function

$$\phi(\lambda) = \frac{1}{2k_1(k_2 - k_1\tau)\lambda^2},$$

which is strictly convex and strictly decreasing for $\lambda > \tilde{\lambda}(\mathbf{K}) = 0$. If $k_3 > 0$, we have the relative state-feedback and for $\lambda > \tilde{\lambda}(\mathbf{K})$ the same argument holds (see the appendix).

Example 1.6.6. Consider a network with nodal matrices

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} -\zeta & 1 \end{bmatrix},$$

for $\zeta > 0$ and $\sigma = 0$. We observer that the subsystems have a non minimum-phase input-output transfer function

$$\mathbf{C}(s\,\mathbf{I}_2 - \mathbf{A})^{-1}\mathbf{B} = \frac{s-\zeta}{s^2},$$

where $\zeta > 0$ is the location of the right-hand plane zero. Let us consider the relative state-feedback. We can show that in this case, we have $\mathbf{P}_0 = \text{diag}\,(2\zeta, 0)$. For a disturbance matrix $\mathbf{E} = [\alpha, \beta]^T$, Theorem 1.5.5 gives us the bound

$$\phi(\lambda, \mathbf{K}) > \text{Tr}\left(\mathbf{E}^T\mathbf{P}_0\mathbf{E}\right) = 2\zeta\alpha^2. \tag{1.51}$$

Alternatively, if we use the relative state-feedback, we can show that the corresponding performance function $\phi(\lambda, \mathbf{K})$ is

$$\frac{\alpha^2(k_1 + k_2\zeta)^2\lambda^2 + (2k_1\alpha^2\zeta^2 + 2k_2\alpha\beta\zeta^2 + k_1\beta^2)\lambda + \beta^2\zeta^2}{2k_1k_2\lambda^2},$$

which is strictly convex and decreasing for $\lambda > \tilde{\lambda}(\mathbf{K}) = 0$. Now, for any gain $\mathbf{K}$ with an unbounded stability region

$$\lim_{\lambda\to\infty} \phi(\lambda, \mathbf{K}) = \frac{\alpha^2(k_1 + k_2\zeta)^2}{2k_1k_2} = \alpha^2\frac{(1 + r\zeta)^2}{2r}, \tag{1.52}$$

where $r := k_2/k_1$. By differentiation with respect to r, we find that the right side attains its minimum at $r = 1/\zeta$. Thus

$$\phi(\lambda, \mathbf{K}) > \lim_{\lambda\to\infty} \phi(\lambda, \mathbf{K})|_{k_2/k_1 = 1/\zeta} = 2\zeta\alpha^2, \tag{1.53}$$

which is the same bound as (1.51). The bound on the performance function scales with the magnitude of the right-hand plane zero at ζ. One inspects that if the disturbance enters

the subsystem from the same channel as the control input, we do not face a fundamental limitation on the performance, because in this case it does not touch the zero dynamics of the subsystems (see [36] for a similar observation in the case of $\mathcal{H}_\infty$-norm).

Example 1.6.7. In this example, first, we consider two different designs for a network of double-integrator agents with measurement noise. Recall that the magnitude of feedback noises is controlled by parameter $\sigma > 0$.

(i) the relative state-feedback without the filtering (i.e., without the decentralized observer): in this case, using $\mathbf{K} = [k_1, k_2]$, we can show that

$$\phi(\lambda, \mathbf{K}) = \frac{1}{k_1 k_2 \lambda^2} + \sigma^2 \frac{k_1^2 + k_2^2}{2k_1 k_2}, \tag{1.54}$$

in which the first term can be recovered from Table 1.1 and the second term appears due to the measurement noises.

(ii) the relative output-feedback on positions with the decentralized observer: in this case

$$\phi(\lambda) = \phi_\xi(\lambda, \mathbf{K}) + \sigma^2 \frac{c_9\lambda^2 + c_{10}\lambda + c_{11}}{c_{12}\lambda^2 + c_{13}\lambda + c_{14}}, \tag{1.55}$$

in which ϕ_ξ is the performance function read from (1.42) and c_9 to c_{14} are polynomials of k_i's and f_i's. For instance, in the case of $k_1 = k_2 = f_1 = f_2 = 1$, this function becomes

$$\phi(\lambda) = \frac{9\lambda^4 + 11\lambda^3 + 9\lambda^2 + 4\lambda + 1}{2(3\lambda^2 + 1)} + \sigma^2 \frac{6\lambda^2 + 5\lambda + 1}{2(3\lambda^2 + 1)}. \tag{1.56}$$

Next, we find estimation function $\psi(\lambda, \mathbf{F})$. We can show that

$$\psi(\lambda, \mathbf{F}) = \frac{1}{2f_1 f_2 \lambda^2} + \frac{\sigma^2}{2}\left(f_1\lambda + \frac{f_2}{f_1}\right). \tag{1.57}$$

The first term is due to the disturbances, while the second term originates from the feedback noises. The transfer matrix $\mathbf{H}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{E}$ is right-invertible and minimum-phase. Hence, as Theorem 1.5.6 suggests, as noise density σ shrinks, one can make the estimation arbitrarily precise by increasing the magnitude of observer gains f_1 and f_2.

1.7 Analysis of Network of Networks

We introduce and analyze a class of networks of networks that are built by a repeated application of control law (1.4). For simplicity of the developments, we neglect the feedback noises (i.e., set $\sigma = 0$). One can show that the same approach works in the presence of those noises as well.

1.7.1 Construction Procedure for Composite Networks

First, we build identical networks using control law (1.4) over graph $\mathcal{G}_1$. We denote the number of nodes of $\mathcal{G}_1$ by m and the order of the state-space realization for each subsystem by n. Moreover, we denote the feedback gain used to build each network by $\mathbf{K}_1 \in \mathbb{R}^{p\times q}$. Let us denote the state of the subsystem j in module or subnetwork i by $x_j^{(i)} \in \mathbb{R}^n$. Similarly, we denote the rest of corresponding variables. The Laplacian matrix corresponding to $\mathcal{G}_1$ is also denoted by $\mathbf{L}_1$. For the subsequent analysis, let us define the following matrices

$$\begin{aligned} &\tilde{\mathbf{A}} := \mathbf{I}_m \otimes \mathbf{A} - \mathbf{L}_1 \otimes \mathbf{B}\mathbf{K}_1\mathbf{H},\ \tilde{\mathbf{B}} := e_m \otimes \mathbf{B}, \\ &\tilde{\mathbf{E}} := \mathbf{I}_m \otimes \mathbf{E},\ \ \tilde{\mathbf{C}} := \mathbf{I}_m \otimes \mathbf{C},\ \tilde{\mathbf{H}} := e_m^T \otimes \mathbf{H}. \end{aligned} \tag{1.58}$$

According to (1.9), the dynamics of the subnetwork i are given by

$$\dot{x}^{(i)} = \tilde{\mathbf{A}}\, x^{(i)} + \tilde{\mathbf{E}}\, \xi^{(i)}, \tag{1.59}$$

where $x^{(i)} := \begin{bmatrix} (x_1^{(i)})^T & \dots & (x_m^{(i)})^T \end{bmatrix}^T \in \mathbb{R}^{mn}$, is the state vector of module i and disturbance vector $\xi^{(i)} \in \mathbb{R}^{mm_1}$ is defined similarly. Without loss of generality, we designate the last node in graph $\mathcal{G}_1$ as the port of the module[2], which corresponds to a subsystem that we can add a term to its control input. This converts (1.59) to new open-loop dynamics

$$\dot{x}^{(i)} = \tilde{\mathbf{A}}\, x^{(i)} + \tilde{\mathbf{B}}\, u^{(i)} + \tilde{\mathbf{E}}\, \xi^{(i)}, \tag{1.60}$$

[2] If we wish to choose another node, we can simply relabel the nodes.

wherein $u^{(i)} \in \mathbb{R}^m$ is a tunable control input to the module. Moreover, we assume that two modules can become interconnected only through their port nodes. Then, the only variable that module i can use for relative feedback is the output variable for the port node, which is denoted by

$$y^{(i)} = \tilde{\mathbf{H}} x^{(i)} = \mathbf{H} x_m^{(i)}. \tag{1.61}$$

We collect N instances of these networks with dynamics (1.60) and feedback variables (1.61) to construct a composite network. Therefore, the subsystems equivalent to S_i in (1.3) for this network design are

$$S^{(i)} : \begin{cases} \dot{x}^{(i)} &= \tilde{\mathbf{A}} x^{(i)} + \tilde{\mathbf{B}} u^{(i)} + \tilde{\mathbf{E}} \xi^{(i)} \\ y^{(i)} &= \tilde{\mathbf{H}} x^{(i)} \\ z^{(i)} &= \tilde{\mathbf{C}} x^{(i)} \end{cases}, \tag{1.62}$$

with the structured matrices defined by (1.58). Now, we build a modular network by application of control law (1.4) with N modules (or subnetworks) connected over a higher level graph $\mathcal{G}_2$ with feedback $\mathbf{K}_2 \in \mathbb{R}^{p \times q}$. [3] If $\{i, j\} \in \mathcal{E}_2$ has a weight denoted by b_{ij}, then the application of control law (1.4) will be

$$u^{(i)} = -\mathbf{K}_2 \sum_{\{i,j\} \in \mathcal{E}_2} b_{ij} \left(y_m^{(i)} - y_m^{(j)} \right). \tag{1.63}$$

We have N modules and each one consists of m subsystems. Therefore, the consensus output of the entire network should be

$$\nu_{\mathrm{nn}}(t) := (\mathbf{M}_{Nm} \otimes \mathbf{C}) x(t). \tag{1.64}$$

Then, we set its steady-state variance as the performance measure

$$\rho_{\mathrm{nn}}(\mathbf{L}_2, \mathbf{K}_2) \;:=\; \lim_{t \to \infty} \mathbb{E} \left\{ \| \nu_{\mathrm{nn}}(t) \|_2^2 \right\}, \tag{1.65}$$

[3] Feedback gains $\mathbf{K}_1$ and $\mathbf{K}_2$ are matrices of the same dimension because we have chosen one node as the port of a subnetwork.

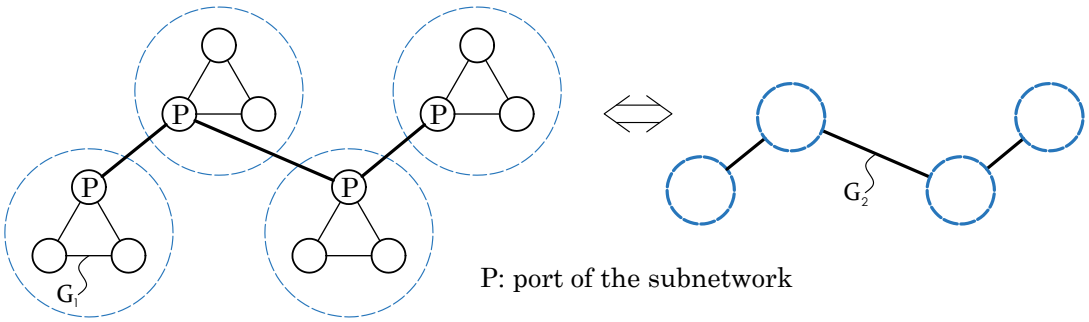

Figure 1.1: An illustration of the proposed model for a network of networks, where the subnetworks over graph $\mathcal{G}_1$ are interconnected via their port nodes (designated with letter **P**) over graph $\mathcal{G}_2$.

where $\mathbf{L}_2$ is the graph Laplacian of $\mathcal{G}_2$.

In Fig. 1.1, we illustrate this composite structure using an example: we have four modules and inside each of them, three subsystems are interconnected over graph $\mathcal{G}_1$, in this case a complete graph. These subnetworks are then connected via their ports over another graph $\mathcal{G}_2$, which in this case is a path graph.

Interpretation of Construction: Let say modules (or subnetworks) i and j are connected, thus $\{i,j\} \in \mathcal{E}_2$. Then, the ports of these two modules will have access to the relative difference of their feedback output $y_m^{(i)} - y_m^{(j)}$ and will reflect this feedback term in their control input. Mathematically speaking, the input to the port node[4] in module i is

$$\begin{aligned} u_m^{(i)} = &- \mathbf{K}_1 \sum_{\{m,k\}\in\mathcal{E}_1} a_{mk} \left(y_m^{(i)} - y_k^{(i)} \right) \\ &- \mathbf{K}_2 \sum_{\{i,j\}\in\mathcal{E}_2} b_{ij} \left(y_m^{(i)} - y_m^{(j)} \right). \end{aligned} \tag{1.66}$$

The first term is due to initial application of control law (1.4) over $\mathcal{G}_1$ with an edge set $\mathcal{E}_1$, while the second term is $u^{(i)}$ from (1.63) based on the composite network design over $\mathcal{G}_2$.

1.7.2 Stability and Performance of Composite Networks

Theorem 1.7.1. *Consider a dynamical network over graph $\mathcal{G}_1$ with a bounded performance measure $\rho(\mathbf{L}_1, \mathbf{K}_1)$. Suppose that in Proposition 1 and Theorem 1 we apply control law (1.4)*

[4] Recall that the port node is arbitrary chosen or labeled to be number m.

on systems $S^{(i)}$ defined in (1.62) over $\mathcal{G}_2$ with feedback gain $\mathbf{K}_2$. The resulting composite network reaches consensus if and only if $\tilde{\mathbf{A}} - \lambda_i(\mathbf{L}_2)\tilde{\mathbf{B}}\mathbf{K}_2\tilde{\mathbf{H}}$ is Hurwitz for nonzero eigenvalues of $\mathbf{L}_2$. Moreover, if $\phi_{\rm nn}(\lambda, \mathbf{K})$ is the performance function derived from Theorem 1 for subsystems $S^{(i)}$ defined in (1.62), then

$$\rho_{\rm nn}(\mathbf{L}_2, \mathbf{K}_2) = \rho(\mathbf{L}_1, \mathbf{K}_1) + \sum_{i=2}^{N} \phi_{\rm nn}\left(\lambda_i(\mathbf{L}_2), \mathbf{K}_2\right). \tag{1.67}$$

The significance of this result is that for a fixed module graph $\mathcal{G}_1$ with Laplacian $\mathbf{L}_1$ and $\mathbf{K}_1$, the value of $\rho(\mathbf{L}_1, \mathbf{K}_1)$ and the form of composite performance function $\phi_{\rm nn}$ are fixed. Thus, we can quantify the role of higher level graph $\mathcal{G}_2$ and feedback gain $\mathbf{K}_2$ in the performance of the composite network by looking at the second term. The extra term compared to Theorem 1.4.2 appears because

$$\nu_{\rm nn} = \left(\mathbf{M}_{Nm} \otimes \mathbf{C}\right) x \neq \left(\mathbf{M}_N \otimes \tilde{\mathbf{C}}\right) x, \tag{1.68}$$

where the right-hand side is the output that would have resulted in an expression of form (1.12).

Remark 7. If a subnetwork is one subsystem, then (1.67) reduces to (1.12), since each subsystem as a network satisfies $\rho(\mathbf{L}_1, \mathbf{K}_1) = 0$.

1.7.3 Minimum Connectivity Design For Composite Networks

We show that if $\tilde{\lambda}(\mathbf{K}_1) < \infty$, then there exists a simple choice for $\mathbf{K}_2$ such that it has also an unbounded stability region in terms of the eigenvalues of higher level Laplacian $\mathbf{L}_2$; i.e., $\tilde{\lambda}(\mathbf{K}_2)$ exists and if $\lambda > \tilde{\lambda}(\mathbf{K}_2)$ then $\tilde{\mathbf{A}} - \lambda\tilde{\mathbf{B}}\mathbf{K}_2\tilde{\mathbf{H}}$ is Hurwitz. This would remedy the concerns about possible complexities in the design of $\mathbf{K}_2$ over graph $\mathcal{G}_2$.

Theorem 1.7.2. *Suppose that the subnetworks are built over any graph $\mathcal{G}_1$ and feedback gain $\mathbf{K}_1$, which has an unbounded stability region. For any $\alpha > 0$, let us choose the feedback gain of the composite network to be $\mathbf{K}_2 = \alpha\mathbf{K}_1$. Then, $\mathbf{K}_2$ has an unbounded stability region with respect to the eigenvalues of higher level Laplacian $\mathbf{L}_2$.*

This result is simplified if $\tilde{\lambda}(\mathbf{K}_1) = 0$.

Dynamics	Performance Function $\phi_{\text{nn}}(\lambda)$
single-integrator	$\dfrac{2(m-1)\lambda + m^2}{2mk\lambda}$
double-integrator	$\dfrac{(m-1)(m+2)\lambda^2 + 2m^2(m-1)\lambda + m^4}{2m^2k_1k_2\lambda^2}$

Table 1.2: Performance functions for a composite network with complete graph subnetworks of single and double-integrator agents. Each module has m nodes and the feedback gains are assumed to be identical over both graphs (see Example 1.7.5).

Corollary 1.7.3. *Suppose that the subnetworks of the network of networks are built with* $\mathbf{K}_1$*, which induces* $\tilde{\lambda}(\mathbf{K}_1) = 0$*. Let us choose* $\mathbf{K}_2 = \alpha\mathbf{K}_1$ *for some* $\alpha > 0$ *in the design of the described composite networks. Then, higher level feedback gain* $\mathbf{K}_2$ *satisfies* $\tilde{\lambda}(\mathbf{K}_2) = 0$ *with respect to the eigenvalues of higher level Laplacian* $\mathbf{L}_2$*.*

1.7.4 Examples of Networks of Networks

Example 1.7.4. Consider a modular network with subnetworks of single-integrators over an unweighted path graph $\mathcal{G}_1$ of m nodes, where the last node of the module is its port. We choose $\mathbf{K}_1 = k_1 > 0$, so the open-loop dynamics of the modules before design of the composite network based on (1.60) are

$$\dot{x}^{(i)} = -k_1\mathbf{L}_1x^{(i)} + e_mu^{(i)} + \mathbf{I}_m\xi^{(i)}, \tag{1.69}$$

where $x^{(i)} \in \mathbb{R}^m, \xi^{(i)} \in \mathbb{R}^m$, $u^{(i)} \in \mathbb{R}$, and $\mathbf{L}_1$ is Laplacian of the unweighted path graph over m nodes. Then, choosing $\mathbf{K}_2 = k_2 > 0$, the performance function of the composite network is

$$\phi_{\text{nn}}(\lambda) = \frac{(m(m-1)/2)k_2\lambda + k_1m}{2k_1k_2\lambda}. \tag{1.70}$$

Based on Corollary 1.7.3 in the higher level $\tilde{\lambda}(\mathbf{K}_2) = 0$. Moreover, we inspect that the resulting family of functions is strictly convex and decreasing for $\lambda > 0$ (see the appendix for the details).

Example 1.7.5. Consider a modular network, where each subnetwork consists of subsystems

with the single or double-integrator dynamics. In this case, we set $\mathcal{G}_1$ to be the unweighted complete graph over m nodes (similar to the example illustrated in Fig. 1.1 for the subnetworks with $m = 3$). Therefore, unlike Example 1.7.4, no matter which node is chosen as the port, the subnetwork will be the identical. For element-wise positive feedback gains $\mathbf{K}_1$ and $\mathbf{K}_2$, Corollary 1.7.3 again implies that the minimum connectivity threshold in terms of the eigenvalues of $\mathbf{L}_2$ for both nodal dynamics is zero. Moreover, let $\mathbf{K}_1 = \mathbf{K}_2 = k > 0$ for the single-integrators and $\mathbf{K}_1 = \mathbf{K}_2 = [k_1, k_2] \succ 0$ for the double-integrators (for simplicity). Using the solution to the Lyapunov equation, we find performance function $\phi_{\rm nn}(\lambda)$ for these subnetworks, which are e given in Table 1.2. These functions are strictly convex and decreasing for $\lambda > 0$. As a sanity check, for $m = 2$ the unweighted path and complete graphs coincide and the first formula in Table 1.2 and the result in (1.70) produce identical functions for $k_1 = k_2 = k$ (see the appendix for more details).

1.8 Performance Bounds and Scaling Laws

We look at the cases where combining the information on graph parameters and derived performance functions can give us macroscopic information on the performance measure. The first type of bounds depend upon the convexity of the performance functions, since they are for networks over any connected graph. The second type of bounds relay on specific graph structures. This waives the requirement for additional properties of the performance functions.

1.8.1 Performance Bounds and Scaling

In Sections 1.6 and 1.7, a majority of the derived performance functions are convex. In what follows, we show that this property is useful in derivation of performance bounds. First resul

Theorem 1.8.1. *Consider a network of N subsystems with a performance functions $\phi(\lambda)$ that is convex. The performance measure over an unweighted graph with M edges and*

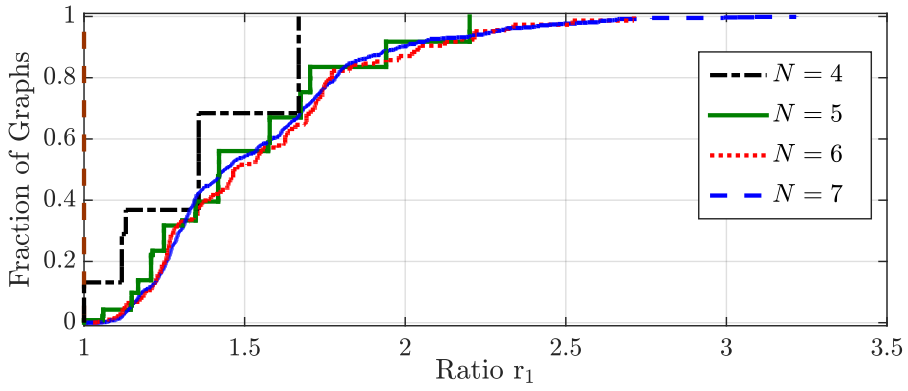

Figure 1.2: The fraction of connected unweighted graphs for which the ratio $\mathfrak{r}_1$ is less than a threshold (see Example 1.8.2)

maximum nodal degree of Δ *is lower-bounded according to*

$$\rho(\mathbf{L},\mathbf{K}) \;\geq\; \phi(1+\Delta)+(N-2)\,\phi\left(\frac{2M-1-\Delta}{N-2}\right), \tag{1.71}$$

where the equality holds if and only if graph $\mathcal{G}$ *is either complete graph or star graph.*

Example 1.8.2. Consider a network with nodal dynamics $\mathfrak{s}_2$, $a_1 = 0$ and $b_0 = b_1 = k_1 = k_2 = a_2 = 1$. For all connected unweighted graphs with 3 to 7 nodes, we do a survey for the ratio of the sides of inequality (1.71), that is

$$\mathfrak{r}_1 := \frac{\rho(\mathbf{L},\mathbf{K})}{\phi(1+\Delta)+(N-2)\phi\left(\dfrac{2M-1-\Delta}{N-2}\right)} \geq 1.$$

The distribution of $\mathfrak{r}_1$ versus N is illustrated in Fig. 1.2. As N increases, it tends to an almost fixed curve (with a growing tail), where about 90% of the graphs induce a ratio $\mathfrak{r}_1$ less than 2. This shows how tight is this bound.

Theorem 1.8.3. *Consider a network of* N *subsystems with a performance functions* $\phi(\lambda)$ *that is convex. The performance measure over any weighted graph with a total weight of* W *is lower-bounded according to*

$$\rho(\mathbf{L},\mathbf{K}) \geq (N-1)\phi\left(\frac{2W}{N-1}\right), \tag{1.72}$$

where the equality holds if and only if the graph is complete and with identical weights.

Theorems 1.8.1 and 1.8.3 give rules of thumb about the best achievable value of the performance measure. To do so, we combine information on the nodal dynamics (through the form of the performance function) and macroscopic level graph information (number of edges or total graph weight).

Corollary 1.8.4. *Under the settings of Theorem 1.8.3, it holds that*

$$\rho(\mathbf{L}, \mathbf{K}) = \Omega\left(N\phi\left(W/N\right)\right). \tag{1.73}$$

Performance-Sparsity Tradeoff: Suppose that $\phi(\lambda)$ is also decreasing. For an unweighted graph, $W = M$. Therefore, we can reorganize the result of Theorem 1.8.3 and write

$$\phi\left(\frac{2M}{N-1}\right) \leq \frac{\rho(\mathbf{L}, \mathbf{K})}{N-1}. \tag{1.74}$$

This result is useful in quantification of the following tradeoff: as the graph of the network becomes sparser, the best attainable value of the performance measure will increase. The following example highlights two specific cases.

Example 1.6.1 (Continued). For networks with subsystems that have $\mathfrak{s}_1$ dynamics, Corollary 1.8.4 implies that over any unweighted graph

$$\rho(\mathbf{L}, \mathbf{K}) = \Omega\left(N^2/M\right). \tag{1.75}$$

For networks with subsystems of $\mathfrak{s}_2$ dynamics we deduce

$$\rho(\mathbf{L}, \mathbf{K}) = \begin{cases} \Omega\left(N^3/M^2\right) & \text{if } b_0 = 0 \\ \Omega\left(N^2/M\right) & \text{if } b_0 \neq 0 \end{cases}. \tag{1.76}$$

For instance, in the special case of $a_1 = a_2 = 0$ for $\mathfrak{s}_2$ we get

$$\frac{b_0^2(N-1)^2}{4k_2M} + \frac{b_1^2(N-1)^3}{8k_1k_2M^2} \leq \rho(\mathbf{L}, \mathbf{K}), \tag{1.77}$$

that clearly reflects the sparsity-performance tradeoff for a consensus network of double-integrators (see [26] for a similar result for single-integrator agents).

1.8.2 Performance Asymptotic over Path and Cycles

We show that we can grasp the asymptotic behavior of $\rho(\mathbf{L}, \mathbf{K})$ over a path or cycle graph by an appropriate integration.

Theorem 1.8.5. *For a network of N subsystems over an unweighted path or cycle graph with a feedback gain $\mathbf{K}$ that satisfies $\tilde{\lambda}(\mathbf{K}) = 0$, it holds that*

$$\rho(\mathbf{L}, \mathbf{K}) = \Theta\left(N\,\Gamma_N\right), \tag{1.78}$$

where Γ_N can be computed using a parametric integral

$$\Gamma_N := \int_{1/N}^{1} \phi\big(2 - 2\cos(\pi x)\big)\; dx. \tag{1.79}$$

Moreover, if $\phi(\lambda)$ is bounded at $\lambda = 0$, then it holds that

$$\lim_{N\to\infty} \frac{N\Gamma_N}{\rho(\mathbf{L}, \mathbf{K})} = 1. \tag{1.80}$$

If $\phi(\lambda)$ is bounded at the origin, we can prove even more.

Corollary 1.8.6. *The performance measure scales similarly with respect to N over unweighted path and cycle graphs. Moreover, if $\phi(\lambda)$ is bounded at 0, the performance measure over the paths and cycles converge to the same value as $N \to \infty$.*

We should emphasize on few points about these results:
(i) Theorem 1.8.5 does not depend on neither convexity nor monotonicity of $\phi(\lambda)$;
(ii) the requirement $\tilde{\lambda}(\mathbf{K}) = 0$ is natural, since as N increases $\lambda_2(\mathbf{L}) = \Theta(1/N^2)$; i.e., it becomes arbitrary small. Otherwise, there exist N_1 such that for $N \geq N_1$, $\lambda_2 < \tilde{\lambda}(\mathbf{K})$.
(iii) This approximation idea has been previously reported, e.g. in [37] it is used for estimation of Estrada index. However, we find the reason for which the approximations find the scaling of the sums, even if $\phi(\lambda)$ is singular at $\lambda = 0$.

Here, we revisit the previous examples and look at bounds and scaling laws for their performance.

Example 1.6.1 (Continued). We apply Theorem 1.8.5 on a network of $\mathfrak{s}_1$ subsystems with $a = 0$ (i.e., for single-integrator agents) over an unweighted path graph and arrive at the asymptotic expression

$$\rho(\mathbf{L}, \mathbf{K}) = \Theta\left(\frac{N^2}{k}\right), \tag{1.81}$$

while if $a > 0$, for $\alpha := a/k$, we have the approximation

$$\rho \sim \frac{N}{2k\sqrt{\alpha(\alpha+4)}}. \tag{1.82}$$

For $\mathfrak{s}_2$ agents with $a_0 = a_1 = 0$, the performance measure satisfies

$$\rho = \Theta\left(\frac{b_0^2 N^2}{2\pi^2 k_2} + \frac{b_1^2 N^4}{6\pi^4 k_1 k_2}\right) := \Theta(h(N, k_1, k_2)). \tag{1.83}$$

Next, for these agents with $b_1 = b_0 = k_1 = k_2 = 1$ over an unweighted path graph of $N = 10, 15, \ldots, 100$ nodes, we investigate the claim of Theorem 1.8.5 by looking at the ratio

$$\mathfrak{r}_2 := \frac{\rho(\mathbf{L}, \mathbf{K})}{h(N, k_1, k_2)}, \tag{1.84}$$

with function h given in (1.83). The result is shown in Fig. 1.3, wherein according to Theorem 1.8.5, $\mathfrak{r}_2$ indeed goes to a constant number.

Example 1.6.4 (Continued). For a network of harmonic oscillators with $\alpha_1 \neq \alpha_2$ over a path graph, Theorem 1.8.5 implies that

$$\rho \sim \frac{N}{2k_1 k_2(\alpha_1 - \alpha_2)}\left(\frac{1}{\sqrt{\alpha_2(\alpha_2+4)}} - \frac{1}{\sqrt{\alpha_1(\alpha_1+4)}}\right).$$

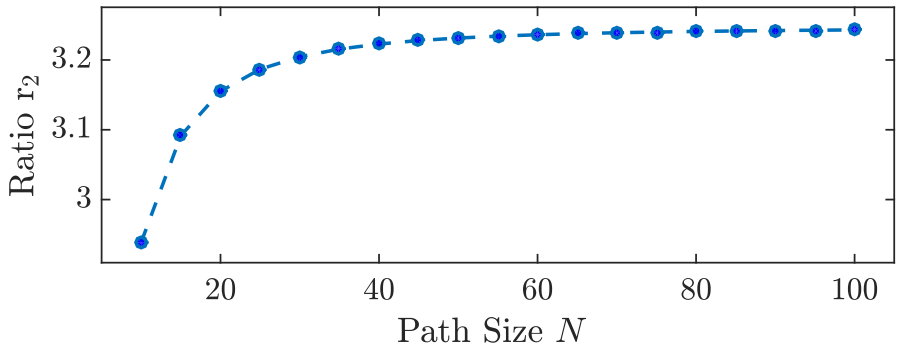

Figure 1.3: Performance asymptotic over paths in continuance of Example 1.6.1.

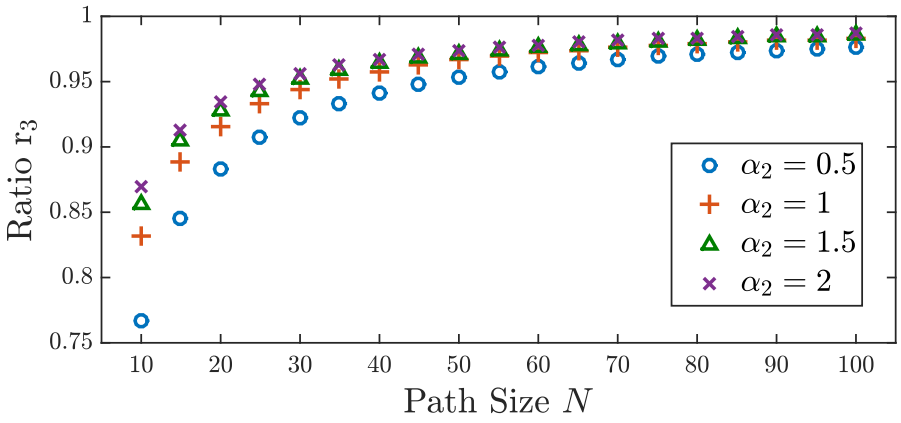

Figure 1.4: Ratio $\mathfrak{r}_3$ for a network of harmonic oscillators over a path graph as the network size grows (see the continuance of Example 1.6.4)

We call the right hand side $f(N, \alpha_1, \alpha_2)$. To empirically examine the gap, we consider

$$\mathfrak{r}_3 := \frac{\rho(\mathbf{L}, \mathbf{K})}{f(N, \alpha_1, \alpha_2)}. \tag{1.85}$$

We set $k_1 = k_2 = 1$, $\alpha_1 = 2\alpha_2$ for $\alpha_2 \in [0.4, 4]$, and vary number of agents N between 10 and 200. Because $\phi(\lambda)$ is bounded at the origin, as N increases the approximation becomes tighter, which is demonstrated in Fig. 1.4.

Example 1.6.5 (Continued) (Platoon over a Path Graph). For a platoon of vehicles over a path graph, Theorem 1.8.5 suggests that $\rho(L, K)$ scales with N^4. Therefore, the $\mathcal{H}_2$-norm scales with N^2. As reported by the authors in [25], a similar scaling law in the case of $\mathcal{H}_\infty$ norm of the network over this topology holds (with an additional leader).

Example 1.6.7 (Continued). We can apply Theorem 1.8.5 to find the scaling for the esti-

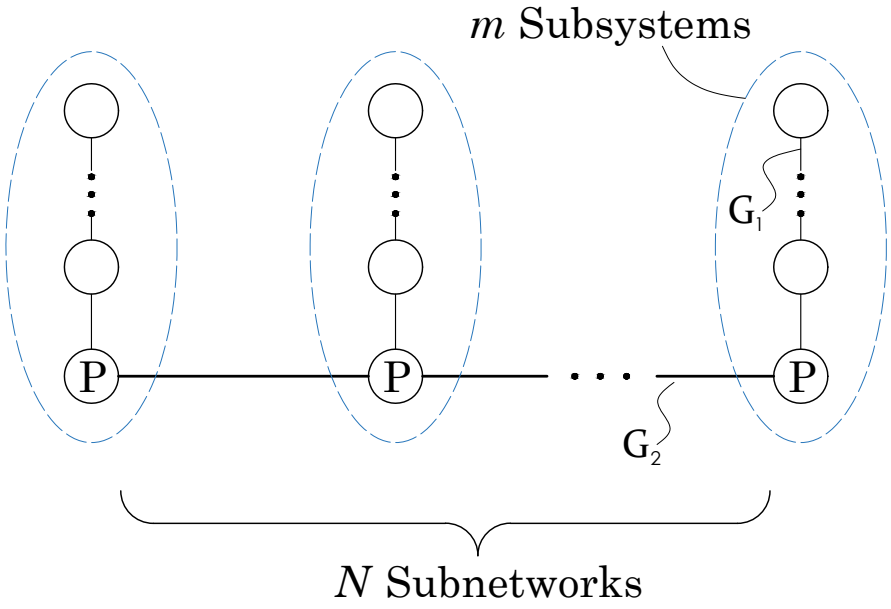

Figure 1.5: Schematic of the composite network whose performance is analyzed in the continuance of Example 1.7.4

mation measure as well. Similar to (1.83), we can show that the estimation measure in a network of double-integrators over a path graph satisfies

$$\mu(\mathbf{L}, \mathbf{F}) = \Theta\left(\frac{N^4}{f_1 f_2} + N\sigma^2\left(f_1 + \frac{f_2}{f_1}\right)\right). \tag{1.86}$$

Example 1.7.4 (Continued). We consider a network of N subnetworks over a path graph, where the subsystems are a network of single integrators, also over a path graph with m subsystems as analyzed in Example 1.7.4. This network is illustrated in Fig. 1.5. From (1.81), we already know that the performance of isolated subnetworks satisfies

$$\rho(\mathbf{L}_1, \mathbf{K}_1) = \Theta\left(m^2/k_1\right). \tag{1.87}$$

Combining (1.87) and Theorem 7, we can show that

$$\rho_{\mathrm{nn}}(\mathbf{L}_2, \mathbf{K}_2) = \Theta\left(\frac{m^2}{k_1} + \frac{m^2 N}{k_1} + \frac{mN^2}{k_2}\right). \tag{1.88}$$

1.9 Application to Formation of Aircraft

Example 1.9.1 (Formation of Aircraft). We consider a linearized model for the dynamics of an aircraft [38] expressed as

$$\dot{X} = \mathbf{A}X + \mathbf{B}\begin{bmatrix} u_1 & u_2 \end{bmatrix}^T + \mathbf{E}\begin{bmatrix} \xi_1 & \xi_2 \end{bmatrix}^T,$$

where $X = \begin{bmatrix} \text{u v } \dot{\theta} \ \theta \text{ x z} \end{bmatrix}^T$ (see the appendix for the numbers). The variable u is the horizontal velocity component from its set point and v is the component normal to that. The pitch angle is denoted by θ. The control inputs u_1 and u_2 are the elevator angle and thrust force, respectively. The scalars ξ_1 and ξ_2 denote the wind velocity in the longitudinal and lateral directions, respectively. We consider the formation shown in Fig. 1.6. Once each vehicle takes into account the relative distances from their neighbors (in computation of the position feedbacks), we can use these dynamics to analyze the performance of this network with the performance output $z = [\alpha \text{x } \beta \text{z}]^T$.

Relative State-Feedback: We use the convex optimization toolbox CVX [39] to find $\mathbf{K}$ for $c = 0.25$ using Theorem 1.5.3. If ξ_1 and ξ_2 have intensity of unity, we get

$$\phi(\lambda) = \alpha^2\phi_1(\lambda) + \beta^2\phi_2(\lambda), \tag{1.89}$$

where $\phi_1(\lambda)$ and $\phi_2(\lambda)$ describe the magnitude of the fluctuations in the formation in x and z directions, respectively. The performance functions are rational functions with the numerator and denominator of order 9. While it is guaranteed to get $\tilde{\lambda}(\mathbf{K}) \in [0, 0.25]$, we have $\tilde{\lambda}(\mathbf{K}) = 0$. In Fig. 1.7, we plot ϕ_1 and ϕ_2, where for larger values of λ, they are different by more than an order of magnitude. The function $\phi_2(\lambda)$ is convex and decreasing, while $\phi_1(\lambda)$ is neither strictly convex nor monotone for $\lambda > 0$. This suggests that properties of these functions in general could be beyond a simple classification.

Observer-Based Relative Output-Feedback: next, we consider the observer-based output-feedback on the last two states of each subsystem (i.e., horizontal and vertical relative positions). For the value of $\mathbf{K}$ we reuse its value from the previous design. We choose

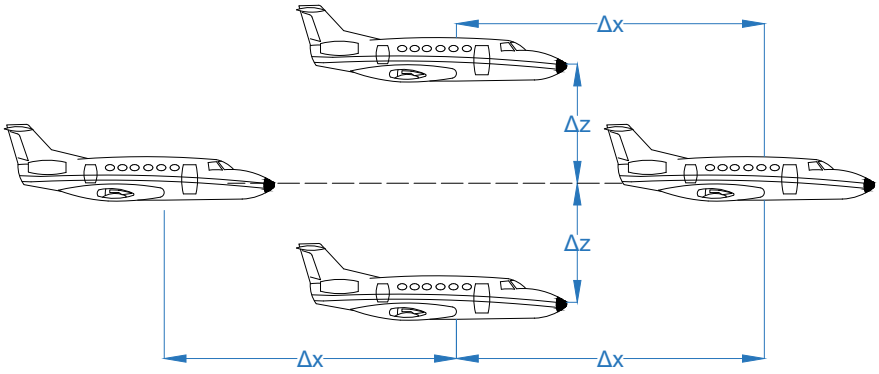

Figure 1.6: The formation of interest in Example 1.9.1

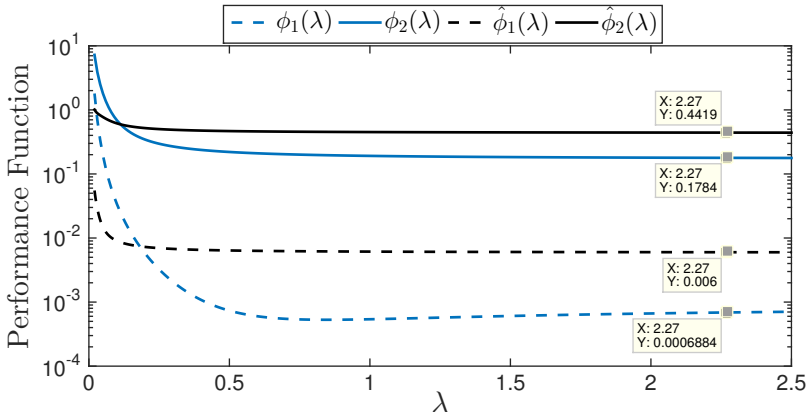

Figure 1.7: The performance functions for Example 1.9.1

observer gain $\mathbf{F}$ for $c = 0.25$ using Theorem 1.5.4 and find that

$$\phi(\lambda) = \alpha^2\hat{\phi}_1(\lambda) + \beta^2\hat{\phi}_2(\lambda), \tag{1.90}$$

where these two functions are also depicted in Fig. 1.7. In this case, $\tilde{\lambda}(\mathbf{F}) = 0$ as well. In Fig. 1.8, we demonstrate two sample longitudinal output plots based on these two designs, where we have 5 planes that are supposed to travel with Δx $= 0.6$. The graph is a path with weights of 4 and identical disturbance samples are fed into the subsystems in two cases. The different level of fluctuations is justifiable upon comparison of the values of ϕ_1 and $\hat{\phi}_1$ in Fig. 1.7.

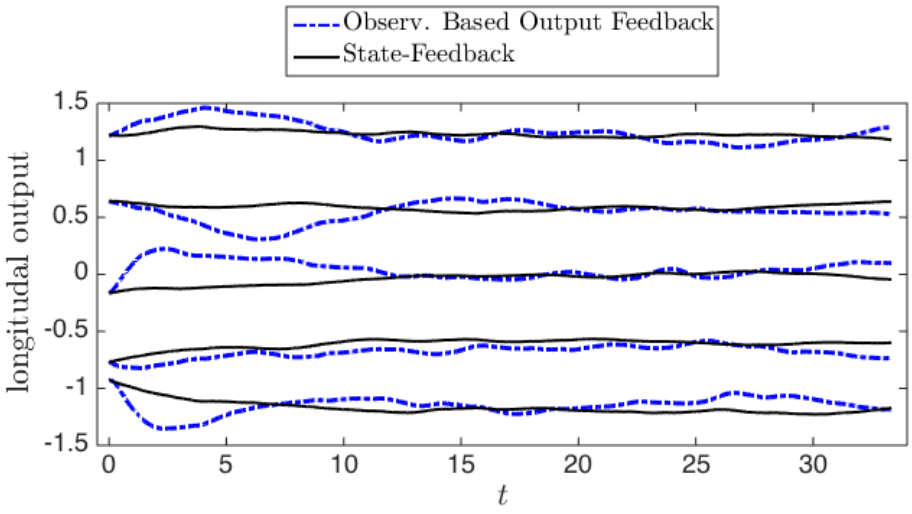

Figure 1.8: The sample outputs based on the designs in Example 1.9.1

1.10 Discussion and Conclusion

We would like to include a a number of remarks:

(i) The spectral expressions for the performance measure can be used to find the optimal values of feedback gain $\mathbf{K}$. In fact, for large networks, solving the Lyapunov equation for the $\mathcal{H}_2$ performance measure once the value of feedback gain is updated could be computationally expensive. Instead, suppose that we find the spectral expressions for the performance measure for a fixed graph. Then, our objective function will be a scalar function of the feedback gain. The resulting problem can be effectively approached using general nonlinear problem methods. This approach is also useful when solving for optimal observer gains $\mathbf{F}$ or feedback gains for composite networks $\mathbf{K}_1$ and $\mathbf{K}_2$ when the graph is fixed. The gains derived from the linear matrix inequalities given in Section 1.5 can be used as a starting point of the optimization procedure.

(ii) Our spectral analysis can be extended to characterize the magnitude of the input signals as well. In fact, we can derive similar spectral expressions for the variance of the control input that is consumed throughout the network in the steady-state, which is given by

$$\rho_u := \lim_{t\to\infty} \mathbb{E}\left\{\|u(t)\|_2^2\right\} \tag{1.91}$$

Then, we can show that

$$\rho_u(\mathbf{L}, \mathbf{K}) = \sum_{i=2}^{N} \phi_u(\lambda, \mathbf{K}). \tag{1.92}$$

for a rational input function $\phi_u(\lambda, \mathbf{K})$, which can be computed by

$$\mu(\lambda, \mathbf{K}) := \mathrm{Tr}\left(\lambda^2 \mathbf{K}\mathbf{H}\mathbf{P}(\lambda, \mathbf{K})\mathbf{H}^T\mathbf{K}^T\right).$$

The map $\mathbf{P}(\lambda, \mathbf{K}) \succ 0$ is the solution to (1.14). For instance, we can show the input functions for networks of single-integrators and double-integrators are given by

$$\phi_u(\lambda, \mathbf{K}) = \frac{k\lambda}{2}, \quad \phi_u(\lambda, \mathbf{K}) = \frac{k_1}{2k_2} + \frac{k_2\lambda}{2},$$

respectively. The developments in this chapter which has to do with the performance functions can be applied to the input functions as well (e.g. asymptotic needed control input over a path).

(iii) In the cases that symbolic evaluation of the performance functions is computationally prohibitive, an alternative option is to conduct regression to estimate the coefficients of these rational performance functions numerically.

Appendix A

Proof of Theorem 1.4.2: Let us define $m = m_1 + m_3$ The transfer matrix from disturbance and noise $[\xi^T, \eta^T]^T$ to consensus output ν can be expressed as

$$\mathbf{G}(s) = (\mathbf{M}_N\mathbf{U} \otimes \mathbf{C})\mathrm{diag}\left(\tilde{\mathbf{G}}_1, \ldots, \tilde{\mathbf{G}}_N\right)(\mathbf{U}^T \otimes \mathbf{I}_m),$$

where $\tilde{\mathbf{G}}_i(s)$ is the transfer matrix from $\left[\chi_i^T, \gamma_i^T\right]^T$ to r_i. This lets us compute the following quantity

$$\mathbf{G}^*(j\omega)\mathbf{G}(j\omega) =$$
$$(\mathbf{U}\otimes\mathbf{I}_m)\text{diag}\left(\tilde{\mathbf{G}}_1^*,\ldots,\tilde{\mathbf{G}}_N^*\right)\left(\mathbf{U}^T\mathbf{M}_N\mathbf{M}_N\mathbf{U}\otimes\mathbf{C}^T\mathbf{C}\right)\text{diag}\left(\tilde{\mathbf{G}}_1,\ldots,\tilde{\mathbf{G}}_N\right)\left(\mathbf{U}^T\otimes\mathbf{I}_m\right).$$

The matrix $\mathbf{U}^T\mathbf{M}_N\mathbf{M}_N\mathbf{U}$ is simply given by

$$\mathbf{U}^T\mathbf{M}_N\mathbf{M}_N\mathbf{U} = \text{diag}(0,1,\ldots,1)\in\mathbb{R}^{N\times N}.$$

Taking $1/(2\pi)\int_{-\infty}^{\infty}\text{Tr}(.)\ d\omega$ from the both sides results in

$$\begin{aligned}\rho(\mathbf{L},\mathbf{K}) =& \frac{1}{2\pi}\int_{-\infty}^{\infty}\text{Tr}\big((\mathbf{U}\otimes\mathbf{I}_m)\\ &\text{diag}\big(0,\tilde{\mathbf{G}}_2^*\mathbf{C}^T\mathbf{C}\tilde{\mathbf{G}}_2,\ldots,\tilde{\mathbf{G}}_N^*\mathbf{C}^T\mathbf{C}\tilde{\mathbf{G}}_N\big)(\mathbf{U}^T\otimes\mathbf{I}_m)\big)\ d\omega.\end{aligned}$$

Due to cyclic property of the trace, the first and last matrix in the trace argument cancel out and we get

$$\frac{1}{2\pi}\int_{-\infty}^{\infty}\text{Tr}\left(\tilde{\mathbf{G}}_i^*\mathbf{C}^T\mathbf{C}\tilde{\mathbf{G}}_i\right)\ d\omega = \left\|\mathbf{C}\tilde{\mathbf{G}}_i(s)\right\|_{\mathcal{H}_2}^2 := \phi(\lambda_i,\mathbf{K}).$$

The last $\mathcal{H}_2$ norm term can be computed using the state-space formulation of systems Σ_i. This will be the Lyapunov equation (1.14) [40]. Therefore, we have managed to prove

$$\rho(\mathbf{L},\mathbf{K}) = \sum_{i=2}^{N}\left\|\mathbf{C}\tilde{\mathbf{G}}_i(s)\right\|_{\mathcal{H}_2}^2 = \sum_{i=2}^{N}\phi(\lambda_i,\mathbf{K}).$$

Next, we prove that $\phi(\lambda,\mathbf{K})$ is a rational function. The Lyapunov equation (1.14) upon vectorization becomes

$$(\mathbf{A}_\lambda\otimes\mathbf{I}_n+\mathbf{I}_n\otimes\mathbf{A}_\lambda)\,\text{vec}(\mathbf{P}) = -\text{vec}\big(\mathbf{E}\mathbf{E}^T+\lambda^2\sigma^2\mathbf{B}\mathbf{K}(\mathbf{B}\mathbf{K})^T\big).$$

Using the Cramer's rule, for $j=1,2,\ldots,n^2$, we may compute the j'th element of vec($\mathbf{P}$)

as

$$\text{vec}(\mathbf{P})_j = \frac{\det((\mathbf{A}_\lambda \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{A}_\lambda)_{-j})}{\det(\mathbf{A}_\lambda \otimes \mathbf{I}_n + \mathbf{I}_n \otimes \mathbf{A}_\lambda)}, \tag{1.93}$$

where $(\mathbf{D})_{-j}$ is the matrix derived by replacing column j of $\mathbf{D}$ with

$$-\text{vec}(\mathbf{E}\mathbf{E}^T + \lambda^2\sigma^2\mathbf{B}\mathbf{K}(\mathbf{B}\mathbf{K})^T).$$

Both numerator and denominator are polynomials of λ with coefficients that are polynomials of the elements of $\mathbf{K}$. Therefore, the same conclusions holds about $\phi(\lambda, \mathbf{K}) = \text{Tr}(\mathbf{C}\mathbf{P}(\lambda, \mathbf{K})\mathbf{C}^T)$.

Appendix B

Proof of Theorem 1.4.3: If we apply control law (1.4) for the new subsystem $\hat{S}_i$ with observer gain $\hat{\mathbf{F}} = [-\mathbf{F}, \mathbf{F}]$ (partitioned based on $\hat{y}_i$), we have the following formula

$$\hat{u}_i = \mathbf{F}\mathbf{H}\Big(\sum_{j\in\mathcal{N}_i} a_{ij}(x_i - x_j) - \sum_{j\in\mathcal{N}_i} a_{ij}(\hat{x}_i - \hat{x}_j)\Big).$$

This means that Σ_i corresponding to the dynamics are

$$\begin{bmatrix}\dot{r}_i \\ \dot{\hat{r}}_i\end{bmatrix} = \begin{bmatrix}\mathbf{A} & -\mathbf{B}\mathbf{K} \\ \lambda\mathbf{F}\mathbf{H} & \mathbf{A} - \mathbf{B}\mathbf{K} - \lambda\mathbf{F}\mathbf{H}\end{bmatrix}\begin{bmatrix}r_i \\ \hat{r}_i\end{bmatrix} + \begin{bmatrix}\mathbf{E} \\ \mathbf{0}\end{bmatrix}\chi_i + \begin{bmatrix}\mathbf{0} \\ -\sigma\lambda_i\mathbf{F}\end{bmatrix}\gamma_i$$

Defining the error as $e_i := r_i - \hat{r}_i$, we get that

$$\begin{bmatrix}\dot{r}_i \\ \dot{e}_i\end{bmatrix} = \begin{bmatrix}\mathbf{A} - \mathbf{B}\mathbf{K} & \mathbf{B}\mathbf{K} \\ \mathbf{0} & \mathbf{A} - \lambda_i\mathbf{F}\mathbf{H}\end{bmatrix}\begin{bmatrix}r_i \\ e_i\end{bmatrix} + \begin{bmatrix}\mathbf{E} & \mathbf{0} \\ \mathbf{E} & \sigma\lambda_i\mathbf{F}\end{bmatrix}\begin{bmatrix}\chi_i \\ \gamma_i\end{bmatrix}. \tag{1.94}$$

The subsystems in the consensus problem have reduced to the familiar decoupled Leunberger observer/regulator form (e.g. see [41]). Therefore, we need to simultaneously have: $\mathbf{A} - \lambda_i\mathbf{F}\mathbf{H}$ for $i = 2, \ldots, N$ and $\mathbf{A} - \mathbf{B}\mathbf{K}$ to be Hurwitz. Then, the corresponding Σ_i is asymptotically stable for $i = 2, \ldots, N$ and the network reaches the consensus.

Appendix C

Proof of Theorem 1.4.4: If we consider the dynamics of e_i in (1.94), it is identical to dynamics of Υ_i in (1.23). The rest of the proof is similar to Theorem 1.4.2 once we replace Σ_i with Υ_i.

Appendix D

Proof of Theorem 1.5.2: The definition of $\tilde{\lambda}(\mathbf{K})$ implies that for all $\lambda > \tilde{\lambda}(\mathbf{K})$, subsystems Σ_2 to Σ_N are asymptotically stable. Therefore, the performance function is bounded. Because $\phi(\lambda)$ is rational, it is analytic everywhere in its domain, including this interval.

Appendix E

Proof of Theorem 1.5.3: First, for a linear time invariant control system the feasibility of the linear matrix inequality and the stabilizability are equivalent [30]. The second part of the claim is a special case of Theorem 11 in [27] with only accounting for the stabilizablity of the subsystems, so we do not repeat the proof in this manuscript. The converse argument holds because if $\tilde{\lambda}(\mathbf{K}) < \infty$, then for $\mathbf{K}^* = 2\tilde{\lambda}(\mathbf{K})\mathbf{K}$, $\mathbf{A} - \mathbf{B}\mathbf{K}^*$ is Hurwitz; i.e., $(\mathbf{A}, \mathbf{B})$ is stabilizable.

Appendix F

Proof of Theorem 1.5.4: Due to duality of between the stabilizability and detectability, if the pair $(\mathbf{A}, \mathbf{H})$ is detectable, then $(\mathbf{A}^T, \mathbf{H}^T)$ is stabilizable. Now, we can use the same argument as Theorem 1.5.3 to complete the proof.

Appendix G

Proof of Theorem 1.5.5: Consider the control system

$$\begin{cases} \dot{x} = \mathbf{A}x + \mathbf{B}u + \mathbf{E}\xi, \\ z = \begin{bmatrix} \mathbf{C}x \\ \epsilon u \end{bmatrix}. \end{cases} \tag{1.95}$$

Saberi et. al. [42] have shown that the minimum value of the $\mathcal{H}_2$ norm for this system is $\gamma^*(\epsilon) = \sqrt{\mathrm{Tr}\,(\mathbf{E}\mathbf{P}_\epsilon\mathbf{E}^T)}$, where $\mathbf{P}_\epsilon$ can be computed as the solution to an algebraic Riccati equation

$$\mathbf{A}^T\mathbf{P}_\epsilon + \mathbf{P}_\epsilon\mathbf{A} + \mathbf{C}^T\mathbf{C} - \frac{1}{\epsilon^2}\mathbf{P}_\epsilon\mathbf{B}\mathbf{B}^T\mathbf{P}_\epsilon = 0. \tag{1.96}$$

Moreover, it has been shown that if $(\mathbf{A}, \mathbf{B})$ is stabilizable and $(\mathbf{A}, \mathbf{C})$ is detectable, then $\mathbf{P}_\epsilon$ converges to zero if and only if the mentioned transfer matrix is right-invertible and minimum-phase. Now, we should note that the performance function is the $\mathcal{H}_2$ norm squared of a system similar to (1.95), expect that we have $\mathbf{B} \to \lambda\mathbf{B}$. Under this modification, the limiting case for $\mathbf{P}_\epsilon$ in Riccati equation (1.96) does not change.

Appendix H

Proof of Theorem 1.7.1: Let us denote orthonormal eigendecompostion of Laplacian $\mathbf{L}_2$ by $\mathbf{L}_2 = \mathbf{U}_2\mathbf{\Lambda}_2\mathbf{U}_2^T$. The decoupled system Σ_i in this case is

$$\dot{r}^{(i)} = \left(\tilde{\mathbf{A}} - \lambda_i(\mathbf{L}_2)\tilde{\mathbf{B}}\mathbf{K}_2\tilde{\mathbf{H}}\right) r^{(i)} + \tilde{\mathbf{E}}\,\chi^{(i)}.$$

Let us call the transfer matrix from $\chi^{(i)}$ to $r^{(i)}$ by $\tilde{\mathbf{G}}^{(i)}$. Then, the transfer matrix from disturbance ξ to performance output ν_{nn} in the case of network of networks can be written as

$$\mathbf{G}_{\mathrm{nn}}(s) = (\mathbf{M}_{Nm} \otimes \mathbf{C})(\mathbf{U}_2 \otimes \mathbf{I}_{mn})\mathrm{diag}(\tilde{\mathbf{G}}^{(1)}, \ldots, \tilde{\mathbf{G}}^{(N)})(\mathbf{U}_2^T \otimes \mathbf{I}_{mm_1}).$$

This lets us compute the following quantity

$$\begin{aligned}(\mathbf{G}_{\rm nn})^*(j\omega)\mathbf{G}_{\rm nn}(j\omega) =&(\mathbf{U}\otimes\mathbf{I}_{m_1})\mathrm{diag}((\tilde{\mathbf{G}}^{(1)})^*,\ldots,(\tilde{\mathbf{G}}^{(N)})^*)\\ &(\mathbf{U}_2^T\otimes\mathbf{I}_{nm})(\mathbf{M}_{Nm}\otimes\mathbf{C}^T\mathbf{C})(\mathbf{U}_2\otimes\mathbf{I}_{nm})\\ &\mathrm{diag}(\tilde{\mathbf{G}}_1,\ldots,\tilde{\mathbf{G}}_N)(\mathbf{U}^T\otimes\mathbf{I}_{m_1}).\end{aligned}$$

We take the trace and move the first two terms of the trace argument to the right to get

$$\begin{aligned}\mathrm{Tr}\,(\mathbf{G}_{\rm nn}^*\mathbf{G}_{\rm nn}) =&\mathrm{Tr}\Big(\mathrm{diag}((\tilde{\mathbf{G}}^{(1)})^*,\ldots,(\tilde{\mathbf{G}}^{(N)})^*)\\ &(\mathbf{U}_2^T\otimes\mathbf{I}_{nm})(\mathbf{M}_{Nm}\otimes\mathbf{C}^T\mathbf{C})(\mathbf{U}_2\otimes\mathbf{I}_{nm})\mathrm{diag}(\tilde{\mathbf{G}}^{(1)},\ldots,\tilde{\mathbf{G}}^{(N)})\Big).\end{aligned}$$

The intermediate term can be simplified according to

$$\left(\mathbf{U}_2^T\otimes\mathbf{I}_{nm}\right)\left(\mathbf{M}_{Nm}\otimes\mathbf{C}^T\mathbf{C}\right)\left(\mathbf{U}_2\otimes\mathbf{I}_{nm}\right)=\mathrm{diag}\Big(\mathbf{M}_m,\underbrace{\mathbf{I}_m,\ldots\mathbf{I}_m}_{N-1\text{ times}}\Big)\otimes\mathbf{C}^T\mathbf{C}.$$

Hence, we can further write

$$\begin{aligned}\mathrm{Tr}\,(\mathbf{G}_{\rm nn}^*\mathbf{G}_{\rm nn}) =&\,\mathrm{Tr}\Big((\tilde{\mathbf{G}}^{(1)})^*(\mathbf{M}_m\otimes\mathbf{C}^T)(\mathbf{M}_m\otimes\mathbf{C})\tilde{\mathbf{G}}^{(1)}\Big)\\ &+\sum_{i=2}^{N}\mathrm{Tr}\Big((\tilde{\mathbf{G}}^{(i)})^*(\mathbf{I}_m\otimes\mathbf{C}^T)(\mathbf{I}_m\otimes\mathbf{C})\tilde{\mathbf{G}}^{(i)}\Big).\end{aligned}$$

If we take the map $1/(2\pi)\int_{-\infty}^{\infty}\mathrm{Tr}(.)\,d\omega$ from the sides

$$\rho_{\rm nn}(\mathbf{L}_2,\mathbf{K}_2)=\left\|(\mathbf{M}_m\otimes\mathbf{C})\tilde{\mathbf{G}}^{(1)}\right\|_{\mathcal{H}_2}^2+\sum_{i=2}^{N}\left\|(\mathbf{I}_m\otimes\mathbf{C})\tilde{\mathbf{G}}^{(i)}\right\|_{\mathcal{H}_2}^2.$$

For $i=1$, $\lambda_1(\mathbf{L}_2)=0$ and the system Σ_1 will have a transfer matrix from the disturbance to output $(\mathbf{M}_m\otimes\mathbf{C})\tilde{\mathbf{G}}^{(1)}$. Moreover, in this case Σ_1 has the closed-loop dynamics of the subnetworks. Therefore, we inspect that

$$\left\|(\mathbf{M}_m\otimes\mathbf{C})\tilde{\mathbf{G}}^{(1)}\right\|_{\mathcal{H}_2}^2=\rho(\mathbf{L}_1,\mathbf{K}_1).$$

Additionally, we observe that for $i = 2, \ldots, N$, we have

$$\left\| (\mathbf{I}_m \otimes \mathbf{C}) \tilde{\mathbf{G}}^{(i)} \right\|_{\mathcal{H}_2}^2 := \phi_{\text{nn}}(\lambda_i(\mathbf{L}_2), \mathbf{K}_2),$$

provided that ϕ_{nn} is the performance function computed using matrices in (1.58).

Appendix I

Proof of Theorem 1.7.2: If $\mathbf{K}_2 = \alpha \mathbf{K}_1$, then we can consider the network of network to be a single network with feedback gain K_1, over a graph $\mathcal{G}_3 = \mathcal{G}_1 \cup \mathcal{G}_2$. The weights of links in $\mathcal{G}_1$ are preserved, while the weights of the links in $\mathcal{G}_2$ are scaled by α. Hence, if we increase the weights in the higher level network (equivalently, the eigenvalues of $\mathbf{L}_2$), at some point the second smallest eigenvalue of equivalent Laplacian $\mathbf{L}_3$ will pass $\tilde{\lambda}(\mathbf{K}_1) < \infty$.

Appendix J

Proof of Corollary 1.7.3: Following the same lines as in the proof of Theorem 1.7.2, if the minimum connectivity threshold is zero, for any choice of $\mathbf{K}_2 = \alpha \mathbf{K}_1$, the network with a single equivalent graph and feedback gain $\mathbf{K}_1$ has zero minimum connectivity threshold, while the equivalent Laplacian would always have a nonzero λ_2. Hence, the connectivity threshold in terms of the eigenvalues of $\mathbf{L}_2$ is zero as well.

Appendix K

Proof of Theorem 1.8.1: First we proof an inequality that is an extension of one in [43] in the case of $f(\lambda) = \lambda^\alpha$ for $\alpha \notin [0, 1]$. For a continuously differentiable convex function $f(x)$ and a Laplacian $\mathbf{L}$ with M edges and maximum degree Δ, we show that

$$\sum_{i=2}^{N} f(\lambda_i) \geq f(1 + \Delta) + (N - 2) f\left(\frac{2M - 1 - \Delta}{N - 2} \right), \tag{1.97}$$

and the equality holds if and only if $\mathcal{G}$ is complete or star. The steps provided in the proof of this lemma are essentially the same steps reported for Theorem 3 in [43] (only for power

functions). Since f is convex and continuous, we use Jensen's inequality to write

$$f\left(\frac{1}{N-2}\sum_{i=2}^{N-1}\lambda_i\right) \leq \frac{1}{N-2}\sum_{i=2}^{N-1} f\left(\lambda_i\right),$$

where if the function $f(\lambda)$ is not affine, then the equality holds if and only if $\lambda_1 = \cdots = \lambda_{N-1}$. This implies we can write

$$\begin{aligned}\sum_{i=2}^{N} f(\lambda_i) &\geq f\left(\lambda_N\right) + (N-2)f\left(\frac{1}{N-2}\sum_{i=2}^{N-1}\lambda_i\right) \\ &= f\left(\lambda_N\right) + (N-2)f\left(\frac{2M-\lambda_N}{N-2}\right) := s(\lambda_N),\end{aligned}$$

where the auxiliary function $s(x)$ is defined as

$$s(x) := f\left(x\right) + (N-2)f\left(\frac{2M-x}{N-2}\right). \tag{1.98}$$

Because $f(x)$ is continuously differentiable, so is $s(x)$ and

$$s'(x) = f'\left(x\right) - f'\left((2M-x)/(N-2)\right).$$

The function $f(x)$ is convex, thus $f'(x)$ is nondecreasing. Then, $s(x)$ is strictly increasing, since for any $x \geq 2M/(N-1)$

$$s'(x) \geq f'\left(\frac{2M}{N-1}\right) - f'\left(\frac{2M-2M/(N-1)}{N-2}\right) \geq f'\left(\frac{2M}{N-1}\right) - f'\left(\frac{2M}{N-2}\right) > 0,$$

In an unweighted graph, $\lambda_N \geq 1+\Delta \geq 2M/(N-1)$ (see [43] and also [44]). Therefore,

$$\sum_{i=2}^{N} f(\lambda_i) \geq s(\lambda_N) \geq s(1+\Delta), \tag{1.99}$$

which proves (1.97). Applying this on a convex $\phi(\lambda)$, (1.71) is followed. The equality holds if and only if $\lambda_2 = \cdots = \lambda_{N-1}$ and $\lambda_N = 1+\Delta$, which happens if and only if $\mathcal{G}$ is either complete or star (again, see both [43] and [44]).

Appendix L

Proof of Theorem 1.8.3: Consider any convex function f. We start from Jensen's inequality in the form of

$$f\left(\frac{1}{N-1}\sum_{i=2}^{N}\lambda_i\right) \leq \frac{1}{N}\sum_{i=2}^{N} f\left(\lambda_i\right).$$

Because the eigenvalues sum to $2W$, replacing f with ϕ gives us the inequality (1.72). The equality holds if and only if $\lambda_2 = \cdots = \lambda_N$ that happens if and only if $\mathcal{G}$ is complete graph with identical weights.

Appendix M

Proof of Theorem 1.8.5: For an unweighted path graph

$$\lambda_i = 2 - 2\cos(\pi(i-1)/N), \text{ for } i = 1, \ldots, N$$

We define the equidistant partition of interval $[1/N, 1]$ as

$$\mathcal{P} = \bigcup_{i=1}^{N-1} [i/N, (i+1)/N] := \bigcup_{i=1}^{N-1} \mathcal{P}_i(N).$$

We define the following quantities for each interval:

$$\overline{\phi}_{i,N} := \max_{x\in\mathcal{P}_i(N)} \phi(2 - 2\cos(\pi x)),$$

$$\phi_{i,N} := \phi(2 - 2\cos(\pi i/N)),$$

$$\underline{\phi}_{i,N} := \min_{x\in\mathcal{P}_i(N)} \phi(2 - 2\cos(\pi x)).$$

They induce the following summations

$$\overline{S}_N = \sum_{i=1}^{N-1} \overline{\phi}_{i,N},\ S_N = \sum_{i=1}^{N-1} \phi_{i,N},\ \underline{S}_N = \sum_{i=1}^{N-1} \underline{\phi}_{i,N},$$

where $\rho(\mathbf{L}, \mathbf{K}) = S_N$. These sums imply the natural ordering

$$\overline{S}_N \leq S_N \leq \underline{S}_N.$$

Moreover, compared to Γ_N, we observe that

$$\underline{S}_N \cdot 1/N \leq \Gamma_N \leq \overline{S}_N \cdot 1/N.$$

We bring a lemma whose proof is given in the next appendix.

Lemma 1.10.1. *For a rational function $\phi(\lambda)$, $\overline{\phi}_{i,N}/\underline{\phi}_{i,N} \leq \delta_\phi$, uniformly over i and N for some $\delta_\phi > 0$ depending on ϕ.*

Applying Lemma 1.10.1, we find that

$$1 \leq \min_{i=1,\ldots,N-1} \frac{\phi_{i,N}}{\underline{\phi}_{i,N}} \leq \frac{\overline{S}_N}{\underline{S}_N} \leq \max_{i=1,\ldots,N-1} \frac{\phi_{i,N}}{\underline{\phi}_{i,N}} \leq \delta_\phi.$$

This means that Γ_N and S_N are both bounded according to

$$1 \leq \frac{N\Gamma_N}{\underline{S}_N} \leq \delta_\phi, \; 1 \leq \frac{S_N}{\underline{S}_N} \leq \delta_\phi.$$

If we combine these two inequalities, we find that

$$\frac{1}{\delta_\phi} \leq \frac{N\Gamma_N}{S_N} \leq \delta_\phi \Rightarrow S_N = \Theta(N\Gamma_N).$$

If $\phi(\lambda)$ is bounded, then one deduces that

$$\lim_{N\to\infty} \overline{\phi}_{i,N}/\underline{\phi}_{i,N} = 1 \Rightarrow \lim_{N\to\infty} \frac{\overline{S}_N}{\underline{S}_N} = 1.$$

Therefore, we can write the following two inequalities

$$\lim_{N\to\infty} \frac{N\Gamma_N}{S_N} = \lim_{N\to\infty} \frac{N\Gamma_N/\underline{S}_N}{S_N/\overline{S}_N} \leq \lim_{N\to\infty} \frac{\overline{S}_N/\underline{S}_N}{\underline{S}_N/\overline{S}_N} = 1$$

$$\lim_{N\to\infty}\frac{N\Gamma_N}{S_N}=\lim_{N\to\infty}\frac{N\Gamma_N/\overline{S}_N}{S_N/\underline{S}_N}\geq\lim_{N\to\infty}\frac{\underline{S}_N/\overline{S}_N}{\overline{S}_N/\underline{S}_N}=1.$$

Thus, we can write

$$\lim_{N\to\infty}\frac{N\Gamma_N}{S_N}=1.$$

For unweighted cycle graphs, the Laplacian eigenvalues are

$$\lambda_i=2-2\cos\left(\frac{2\pi(i-1)}{N}\right),\ \text{for } i=1,\ldots,N.$$

Without loss of generality, for deriving the scaling purposes, we may assume that N is *odd*. Then, we will have $(N-1)/2$ distinct values for the eigenvalues of $\mathbf{L}$, where each value is repeated exactly twice. Moreover, we can write

$$\rho(\mathbf{L},\mathbf{K})=2\sum_{i=1}^{(N-1)/2}\phi\left(2-2\cos\left(\frac{2\pi i}{N}\right)\right).$$

This time, we need to partition $[1/N,1/2]$ and proceed with identical steps to find out that

$$2\int_{1/N}^{1/2}\phi\left(2-2\cos\left(2\pi x\right)\right)\ dx,$$

does the same job in the case of cycle graphs. If we replace $2x\to x$, the factor 2 is cancelled. This means that (1.78) and (1.80) upon replacement of Γ_N with Ψ_N are achieved, where

$$\Psi_N:=\int_{2/N}^{1}\phi\big(2-2\cos(\pi x)\big)\ dx; \tag{1.100}$$

Since $\Gamma_N=\Theta(\Psi_N)$, the same conclusion apply for the networks over cycle graphs as well.

Appendix N

Proof of Lemma 1.10.1: Let us define

$$\chi:=\log(2-2\cos(\pi x)),$$

and also

$$\varphi(\chi) := \log(\phi(2 - 2\cos(\pi x))).$$

Denote the order of the pole of $\phi(x)$ at $x = 0$ by $\alpha \in \mathbb{Z}_+$. This means that we can decompose φ according to

$$\begin{aligned}\varphi(\chi) &= \log\left(\frac{1}{(2 - 2\cos(\pi x))^{\alpha}}\hat{\varphi}(2 - 2\cos(\pi x))\right) \\ &= \log\left(\hat{\varphi}(2 - 2\cos(\pi x))\right) - \alpha \log(2 - 2\cos(\pi x)),\end{aligned}$$

for some strictly positive function $\hat{\varphi}(.)$ that is bounded and rational. We can write this in terms of χ as follows

$$\varphi(\chi) = \log\left(\hat{\varphi}(\exp(\chi))\right) - \alpha\chi.$$

In the interval of interest, $\log(.)$, $\hat{\varphi}(.)$ (a bounded and positive rational function) and $\exp(.)$ are all Lipschitz continuous, so is their composition $\log\left(\hat{\varphi}(\exp(\chi))\right)$. This implies that $\varphi(\chi)$ is Lipschitz-continuous. Hence, for $\chi_1 = \log(2 - 2\cos(\pi x_1))$ and $\chi_2 = \log(2 - 2\cos(\pi x_2))$, the corresponding Lipschitz continuity inequality will be

$$|\varphi(\chi_2) - \varphi(\chi_1)| \leq \delta_{\varphi}|\chi_2 - \chi_1|,$$

for some Lipschitz constant $\delta_{\varphi} \geq 0$. This is equivalent to

$$\left|\log\left(\frac{\phi(x_2)}{\phi(x_1)}\right)\right| \leq \delta_{\varphi}\left|\log\left(\frac{2 - 2\cos(\pi x_2)}{2 - 2\cos(\pi x_1)}\right)\right| = \left|\log\left(\frac{2 - 2\cos(\pi x_2)}{2 - 2\cos(\pi x_1)}\right)^{\delta_{\varphi}}\right|.$$

This means that alternatively we may write

$$\max\left(\frac{\phi(x_2)}{\phi(x_1)}, \frac{\phi(x_1)}{\phi(x_2)}\right) \leq \max\left(\left(\frac{2 - 2\cos(\pi x_2)}{2 - 2\cos(\pi x_1)}\right)^{\delta_{\varphi}}, \left(\frac{2 - 2\cos(\pi x_1)}{2 - 2\cos(\pi x_2)}\right)^{\delta_{\varphi}}\right). \tag{1.101}$$

Let us define

$$h_N(x) := \frac{2 - 2\cos\left(\pi\left(x + 1/N\right)\right)}{2 - 2\cos(\pi x)}.$$

Since $2-2\cos(\pi x)$ is increasing in $\mathcal{P}_i(N)$, for $x_1, x_2 \in \mathcal{P}_i(N)$

$$\frac{2-2\cos(\pi x_2)}{2-2\cos(\pi x_1)} \leq h_N(i/N).$$

Moreover, one can find that $h_N(x)$ is decreasing for any $x \in [1/N, (N-1)/N]$. Hence, we can write

$$\frac{2-2\cos(\pi x_2)}{2-2\cos(\pi x_1)} \leq h_N(i/N) \leq h_N(1/N). \tag{1.102}$$

We can see note that the right hand side of (1.102) is

$$h_N(1/N) = \frac{2-2\cos\left(2\pi/N\right)}{2-2\cos(\pi/N)}.$$

Computing its *formal* derivative with respect to N, we get

$$\frac{dh_N(1/N)}{dN} = \frac{2\pi\sin\left(\pi/N\right)}{N^2} > 0.$$

Thus, its supremum should be evaluated based on the limit

$$\frac{2-2\cos(\pi x_2)}{2-2\cos(\pi x_1)} \leq \lim_{N\to\infty} h_N(1/N) = 4. \tag{1.103}$$

Let us take the maximum of the left hand side of (1.101) over $x_1, x_2 \in \mathcal{P}_i(N)$ and combine it with (1.103). We conclude that

$$\frac{\overline{\phi}_{i,N}}{\underline{\phi}_{i,N}} \leq 4^{\delta_\varphi} := \delta_\phi,$$

which completes the proof.

Appendix O

Proof of Corollary 1.8.6: Because Γ_N and Ψ_N scale similarly with respect to N, the first part of the claim follows. If the performance function is bounded, then both of them become

the same quantity, because we can replace the lower bound of these two integrals with their common limit of 0.

Appendix P

Proof of Result about Input Measures in Section 1.10: We can see that

$$\|u\|_2^2 = \mathrm{Tr}\left(uu^T\right) = (\mathbf{L}\otimes\mathbf{KH})xx^T(\mathbf{L}\otimes\mathbf{H}^T\mathbf{K}^T).$$

We can further write

$$uu^T = (\mathbf{L}\otimes\mathbf{KH})xx^T(\mathbf{L}\otimes\mathbf{H}^T\mathbf{K}^T).$$

Because $\mathbf{M}_N\mathbf{L} = \mathbf{L}\mathbf{M}_N = \mathbf{L}$, we can write

$$uu^T = (\mathbf{L}\otimes\mathbf{KH})(\mathbf{M}_N x)(\mathbf{M}_N x)^T(\mathbf{L}\otimes\mathbf{H}^T\mathbf{K}^T).$$

If we take the expected value and tend the time to infinity, using the same lines as the proof of Theorem 1, we find that

$$\lim_{t\to\infty}\mathbb{E}\{u(t)u(t)^T\} =$$
$$(\mathbf{L}\otimes\mathbf{KH})(\mathbf{U}\otimes\mathbf{I}_n)\mathrm{diag}(\mathbf{0},\mathbf{P}(\lambda_2,\mathbf{K}),\ldots,\mathbf{P}(\lambda_N,\mathbf{K}))(\mathbf{U}^T\otimes\mathbf{I}_n)(\mathbf{L}\otimes\mathbf{H}^T\mathbf{K}^T).$$

Let us take the trace from the both sides. We find that

$$\rho_u(\mathbf{L},\mathbf{K}) = \mathrm{Tr}\left(\mathrm{diag}(\mathbf{0},\mathbf{P}(\lambda_2,\mathbf{K}),\ldots,\mathbf{P}(\lambda_N,\mathbf{K}))\ (\mathbf{U}^T\mathbf{L}^2\mathbf{U}\otimes\mathbf{H}^T\mathbf{K}^T\mathbf{KH})\right).$$

Because $\mathbf{U}^T\mathbf{L}^2\mathbf{U} = \mathbf{\Lambda}^2$, we conclude that

$$\begin{aligned}\rho_u(\mathbf{L},\mathbf{K}) =& \mathrm{Tr}\left(\mathrm{diag}(\mathbf{0},\lambda_2^2\mathbf{P}(\lambda_2,\mathbf{K})\ \mathbf{H}^T\mathbf{K}^T\mathbf{KH},\ldots,\lambda_N^2\mathbf{P}(\lambda_N,\mathbf{K})\mathbf{H}^T\mathbf{K}^T\mathbf{KH}\right)\\ =& \sum_{i=2}^{N}\mathrm{Tr}\left(\lambda_i^2\mathbf{P}(\lambda_i,\mathbf{K})\mathbf{H}^T\mathbf{K}^T\mathbf{KH}\right).\end{aligned}$$

Let us move **KH** to the left hand side of the trace arguments. The claim is followed. □

Appendix Q

Details of Example 1.6.1: For networks with nodal dynamics $\mathfrak{s}_1$, the noiseless dynamics of Σ_i are

$$\dot{r} = (-a - \lambda_i k)r,$$

that are asymptotically stable if $a - \lambda_i k < 0$. This confirms $\tilde{\lambda}(\mathbf{K}) = \max(-a/k, 0) = 0$. The solution to (1.14) is $\mathbf{P} = 1/2(k\lambda + a)$, which result in the claimed form for ϕ. We can see that for $\lambda > \tilde{\lambda}(\mathbf{K}) = 0$, it is strictly convex and strictly decreasing. The dynamics of the subsystem Σ_i for nodal dynamics $\mathfrak{s}_2$ without disturbance and noise are

$$\dot{r} = \begin{bmatrix} 0 & 1 \\ -a_2 - \lambda k_1 & -a_1\lambda k_2 \end{bmatrix} r.$$

The corresponding characteristic polynomial is

$$p_\lambda(s) = s^2 + (a_1 + k_2\lambda)s + a_2 + k_1\lambda,$$

which is a stable polynomial if and only if $a_1 + k_2\lambda > 0$, and $a_2 + k_1\lambda > 0$, that imply $\tilde{\lambda}(\mathbf{K}) = \max(-a_1/k_2, -a_2/k_1, 0) = 0$. Using (1.14), we get that

$$\mathbf{P}(\lambda, \mathbf{K}) = \operatorname{diag}\left(\frac{1}{2(k_2\lambda + a_1)(k_1\lambda + a_2)}, \frac{1}{2(k_2\lambda + a_1)}\right).$$

Substitution of this matrix into (1.13) gives us the results of the table. Now, noting that

$$\phi(\lambda) = \frac{b_0^2}{2(k_2\lambda + a_1)} + \frac{b_1^2}{2(k_2\lambda + a_1)(k_1\lambda + a_2)},$$

that is sum of two strictly convex and strictly decreasing functions after their negative poles. Those poles are less than or equal to $\tilde{\lambda}(\mathbf{K})$. Hence, the $\phi(\lambda)$ is strictly decreasing and strictly convex in the claimed domain.

$$\frac{1}{2}\begin{bmatrix} \dfrac{k_3\lambda}{(k_1\lambda)(k_2k_3\lambda^2\lambda)-k_1\lambda)} & 0 & * \\ 0 & \dfrac{1}{k_2k_3\lambda^2-k_1\lambda} & 0 \\ \dfrac{-1}{k_2k_3\lambda^2-k_1\lambda} & 0 & \dfrac{k_2\lambda}{k_2k_3\lambda^2-k_1\lambda} \end{bmatrix}$$

Table 1.3: The value of $\mathbf{P}(\lambda, \mathbf{K})$ as the solution of Lyapunov equation for triple integrators.

For the double integrator with observer, note that

$$\mathbf{A}_\lambda = \mathbf{A} - \lambda \mathbf{F}\mathbf{H} = \begin{bmatrix} -f_1\lambda & 1 \\ -f_2\lambda & 0 \end{bmatrix}.$$

Observe that its characteristic polynomial is $p_\lambda(s) = s^2 + f_1\lambda s + f_2\lambda$, which is stable if and only if $f_1, f_2 > 0$.

Details of Example 1.6.3: For the network of triple-integrators, the characteristic polynomial Σ_i is

$$p_\lambda(s) = s^3 + k_3\lambda s^2 + k_2\lambda s + k_1\lambda.$$

The stability requires $k_i > 0$ and

$$(k_2\lambda)(k_3\lambda) - (k_1\lambda) > 0 \Rightarrow \lambda > \frac{k_1}{k_2k_3} \Rightarrow \tilde{\lambda}(\mathbf{K}) = \frac{k_1}{k_2k_3}.$$

The solution to for $\mathbf{P}(\lambda, \mathbf{K})$ from the Lyapunov equation is shown in Table 1.3. The formula for the performance function then is followed by computing $\mathrm{Tr}(\mathbf{C}^T\mathbf{P}\mathbf{C})$.The performance function in this case is strictly decreasing and convex.

Details of Example 1.6.5: The realization of the agents is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1/\tau \end{bmatrix}, \ \mathbf{B} = \mathbf{E} = \begin{bmatrix} 0 \\ 0 \\ 1/\tau \end{bmatrix}, \ \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

The characteristic polynomial of the systems in this case becomes

$$p_\lambda(s) = \tau s^3 + (1 + \lambda k_3)s^2 + \lambda k_2 s + \lambda k_1,$$

(see also [25]). Based on Routh-Hurwitz criteria, for an unbounded stability region, we should impose the following restrictions on $\mathbf{K}$. $k_1, k_2 > 0,$ and $k_3 \geq 0$. Moreover, we need the inequality

$$(1 + \lambda k_3)\lambda k_2 > \tau\lambda k_1 \Rightarrow k_3\lambda > \tau k_1/k_2 - 1, \tag{1.104}$$

from which we may infer the bi-criteria definition for $\tilde{\lambda}(\mathbf{K})$ in (1.49). To find the performance functions, we find $\mathbf{P}(\lambda, \mathbf{K})$ which is

$$\begin{bmatrix} \frac{1}{2k_1}\dfrac{k_3\lambda + 1}{k_2k_3\lambda^3 + (k_2 - k_1\tau)\lambda^2} & 0 & \dfrac{-1}{2(k_2k_3\lambda^2 + (k_2 - k_1\tau)\lambda)} \\ 0 & \dfrac{1}{2(k_2k_3\lambda^2 + (k_2 - k_1\tau)\lambda)} & 0 \\ \dfrac{-1}{2(k_2k_3\lambda^2 + (k_2 - k_1\tau)\lambda)} & 0 & \dfrac{k_2}{2\tau(k_2k_3\lambda + k_2 - k_1\tau)} \end{bmatrix}$$

which lets us compute the performance function. If $k_3 > 0$, the function $\phi(\lambda)$ is a positive combination of

$$f_1(\lambda) = \left(k_3\lambda^2 + \frac{k_2 - k_1\tau}{k_2}\lambda\right)^{-1}$$
$$f_2(\lambda) = \left(k_3\lambda^3 + \frac{k_2 - k_1\tau}{k_2}\lambda^2\right)^{-1}.$$

Moreover, f_1 and f_2 are products of functions that are strictly convex and decreasing for $\lambda > \tilde{\lambda}(\mathbf{K})$ according to

$$f_1(\lambda) = \frac{1}{k_3\lambda} \cdot \frac{1}{\lambda + \dfrac{k_2 - k_1\tau}{k_3k_2}}$$
$$f_2(\lambda) = \frac{1}{k_3\lambda^2} \cdot \frac{1}{\lambda + \dfrac{k_2 - k_1\tau}{k_3k_2}}.$$

Thus, the performance function is this case is also strictly convex and strictly decreasing

for $\lambda > \tilde{\lambda}(\mathbf{K})$.

Details of Example 1.6.6: The input-output transfer function is $H(s) = (s-\zeta)/s^2$ with a right-hand plane zero at $\zeta > 0$. To evaluate $\mathbf{P}_0$, we use a method suggested by [45]. First, we decompose the transfer function according to $H = H_1 H_2$ where the components are

$$H_1(s) = \frac{s-\zeta}{s+\zeta},\ H_2(s) = \frac{s+\zeta}{s^2}.$$

The transfer function H_1 has the balanced realization [5].

$$\hat{\mathbf{A}}_1 = -\zeta,\ \hat{\mathbf{B}} = \sqrt{2\zeta},\ \hat{\mathbf{C}}_1 = -\sqrt{2\zeta},\ \hat{\mathbf{D}}_1 = 1,$$

while H_2 has a stabilizable and detectable realization

$$\hat{\mathbf{A}}_2 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},\ \hat{\mathbf{E}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ \hat{\mathbf{C}}_2 = \begin{bmatrix} \zeta & 1 \end{bmatrix},\ \hat{\mathbf{D}}_2 = 0.$$

Suppose that the factorized realizations have the state vectors X and x, respectively. Then, we can show that they are related based on

$$X = \begin{bmatrix} \sqrt{2\zeta} & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} x := \mathbf{T}x.$$

Then, the reference shows that

$$\mathbf{P}_0 = \mathbf{T}^T \begin{bmatrix} \mathbf{I}_b & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{T}. \tag{1.105}$$

Therefore, we get $\mathbf{P}_0 = \mathrm{diag}\,(2\zeta, 0)$.

[5] A minimal realization of a stable transfer matrix is called balanced if its controllability and observability Gramians are diagonal and equal (such a realization exists for a stable transfer matrix)

$$\frac{1}{2}\begin{bmatrix} \mathbf{J}_{m-1}\otimes\begin{bmatrix}\frac{(m+1)\lambda^2+2m^2\lambda+m^4}{m^2k_1k_2\lambda^2} & 0\\ 0 & \frac{\lambda+m}{mk_2\lambda}\end{bmatrix}+\mathbf{I}_{m-1}\otimes\begin{bmatrix}\frac{1}{m^2k_1k_2} & 0\\ 0 & \frac{1}{mk_2}\end{bmatrix} & * \\ 1_{m-1}^T\otimes\begin{bmatrix}\frac{\lambda+m}{k_1k_2\lambda^2} & 0\\ 0 & \frac{1}{k_2\lambda}\end{bmatrix} & \begin{bmatrix}\frac{m}{k_1k_2\lambda^2} & 0\\ 0 & \frac{1}{k_2\lambda}\end{bmatrix}\end{bmatrix}$$

Table 1.4: The solution to Lyapunov equation $\tilde{\mathbf{P}}$ for complete subnetworks with m double-integrator agents ($*$ implies symmetric element).

Details of Example 1.7.4: We can see that $\tilde{\mathbf{E}}=\tilde{\mathbf{C}}=\mathbf{I}_m$ and

$$\tilde{\mathbf{A}}_\lambda=\begin{bmatrix}-k_1 & k_1 & 0 & 0 & \dots & 0\\ k_1 & -2k_1 & k_1 & 0 & \dots & 0\\ \vdots & \ddots & & \ddots & \ddots & \vdots\\ 0 & \dots & 0 & 0 & k_1 & -k_1-k_2\lambda\end{bmatrix}.$$

The solution to the Lyapunov equation in this case is

$$\tilde{\mathbf{P}}=\begin{bmatrix}\frac{k_1+(m-1)k_2\lambda}{2k_1k_2\lambda} & \frac{k_1+(m-2)k_2\lambda}{2k_1k_2\lambda} & \cdots & \frac{k_1+k_2\lambda}{2k_1k_2\lambda} & \frac{1}{2k_2\lambda}\\ \frac{k_1+(m-2)k_2\lambda}{2k_1k_2\lambda} & \frac{k_1+(m-2)k_2\lambda}{2k_1k_2\lambda} & \cdots & \frac{k_1+k_2\lambda}{2k_1k_2\lambda} & \frac{1}{2k_2\lambda}\\ \vdots & \vdots & \ddots & \frac{k_1+k_2\lambda}{2k_1k_2\lambda} & \frac{1}{2k_2\lambda}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \frac{k_1+k_2\lambda}{2k_1k_2\lambda} & \frac{k_1+k_2\lambda}{2k_1k_2\lambda} & \cdots & \frac{k_1+k_2\lambda}{2k_1k_2\lambda} & \frac{1}{2k_2\lambda}\\ \frac{1}{2k_2\lambda} & \frac{1}{2k_2\lambda} & \cdots & \frac{1}{2k_2\lambda} & \frac{1}{2k_2\lambda}\end{bmatrix}.$$

Now, similar to the previous example, we can see that

$$\begin{aligned}\phi_{\rm nn}=\mathrm{Tr}(\tilde{\mathbf{C}}\tilde{\mathbf{P}}\tilde{\mathbf{C}}^T)=\mathrm{Tr}(\tilde{\mathbf{P}})&=\sum_{i=1}^m\frac{k_1+(i-1)k_2\lambda}{2k_1k_2\lambda}\\ &=\frac{k_1m+k_2\lambda\sum_{i=1}^m(i-1)}{2k_1k_2}=\frac{\frac{m(m-1)}{2}k_2\lambda+k_1m}{2k_1k_2}.\end{aligned}$$

Details of Example 1.7.5: For subnetworks of single-integrators over $\mathcal{G}_1$ that is complete,

$\tilde{\mathbf{E}} = \tilde{\mathbf{C}} = \mathbf{I}_m$ and

$$\tilde{\mathbf{A}}_\lambda = \begin{bmatrix} -(m-1)k & k & \dots & k \\ \vdots & \ddots & & \vdots \\ k & \dots & k & -(m-1)k - k\lambda \end{bmatrix}.$$

We can verify that the solution to the Lyapunov equation is

$$\tilde{\mathbf{P}} = \frac{1}{2k} \begin{bmatrix} \frac{\lambda+m}{m\lambda}\mathbf{J}_{m-1} + \frac{1}{m}\mathbf{I}_{m-1} & \frac{1}{\lambda}1_{m-1} \\ \frac{1}{\lambda}1_{m-1}^T & \frac{1}{\lambda} \end{bmatrix}.$$

Then, we can write

$$\begin{aligned} \phi_{\text{nn}} &= \text{Tr}(\tilde{\mathbf{C}}\tilde{\mathbf{P}}\tilde{\mathbf{C}}^T) = \text{Tr}(\tilde{\mathbf{P}}) \\ &= (m-1)\frac{1}{2k}\left(\frac{\lambda+m}{m\lambda} + \frac{1}{m}\right) + \frac{1}{2k\lambda} = \frac{2(m-1)\lambda + m^2}{2mk\lambda}. \end{aligned}$$

For double-integrators over complete graph modules, similar expressions for the matrices $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$, and $\tilde{\mathbf{C}}$ holds, while the solution to the Lyapunov equation in this case is shown in Table 1.4. Because the output of the double-integrator is on the first state, we can write

$$\begin{aligned} \phi_{\text{nn}} = \text{Tr}(\tilde{\mathbf{C}}\tilde{\mathbf{P}}\tilde{\mathbf{C}}^T) &= \frac{m-1}{2}\big(\frac{(m+1)\lambda^2 + 2m^2\lambda + m^4}{m^2k_1k_2\lambda^2} + \frac{1}{m^2k_1k_2}\big) + \frac{m}{2k_1k_2\lambda^2} \\ &= \frac{(m-1)(m+2)\lambda^2 + 2m^2(m-1)\lambda + m^4}{2m^2k_1k_2\lambda^2}. \end{aligned}$$

This proves the claims in the example.

Details of Continuance of Example 1.6.1: First, we prove that we can replace Γ_N in Theorem 1.8.5 with

$$\Gamma_N = \frac{1}{2\pi}\int_{\pi^2/N^2}^{4} \phi(\lambda)\frac{1}{\sqrt{\lambda - \lambda^2/4}}\, d\lambda.$$

Considering $\lambda = 2 - 2\cos(\pi x)$, we get that

$$d\lambda = 2\pi \sin(\pi x)\, dx = 2\pi\sqrt{1 - \cos^2(\pi x)}\, dx.$$

Given $\lambda = 2 - 2\cos(\pi x)$, $d\lambda = 2\pi\sqrt{\lambda - \lambda^2/4}\, dx$, and

$$\begin{cases} x = 1/N \Rightarrow \lambda = 2 - 2\cos(\pi/N) \sim \pi^2/N^2 \\ x = 1 \Rightarrow \lambda = 2 + 2 = 4 \end{cases}.$$

that are integral limits of interest.

Now, if $a = 0$, for the single integrators

$$\int \frac{1}{\lambda} \frac{1}{\sqrt{\lambda - \lambda^2/4}}\, d\lambda = -\frac{\sqrt{4-\lambda}}{\sqrt{\lambda}},$$

we can compute Γ_N for $\phi(\lambda)$ as

$$\Gamma_N = \frac{1}{2k}\frac{1}{2\pi}\left(\frac{\sqrt{4-\pi^2/N^2}}{\sqrt{\pi^2/N^2}}\right) \sim \frac{N}{2\pi^2 k}.$$

Now, if $a > 0$, then we need the integral

$$\int_0^4 \frac{1}{\lambda + \alpha} \frac{1}{\sqrt{\lambda - \lambda^2/4}}\, d\lambda = \frac{2\pi}{\sqrt{\alpha(\alpha+4)}}.$$

This implies that Γ_N for $\phi(\lambda)$ in this case satisfies

$$\Gamma_N \sim \frac{1}{2k}\frac{1}{2\pi}\frac{2\pi}{\sqrt{\alpha(\alpha+4)}} = \frac{1}{2k\sqrt{\alpha(\alpha+4)}}.$$

For $\mathfrak{s}_2$ agents with $a_0 = a_1 = 0$ we need

$$\int \frac{1}{\lambda^2} \frac{1}{\sqrt{\lambda - \lambda^2/4}}\, d\lambda = -\frac{\sqrt{4-\lambda}(\lambda+2)}{6\lambda\sqrt{\lambda}}.$$

Now, we compute Γ_N for $\phi(\lambda)$ as follows

$$\Gamma_N \sim \frac{1}{2\pi}\frac{b_0^2}{2k_2}\left(\frac{\sqrt{4-\pi^2/N^2}}{\sqrt{\pi^2/N^2}}\right) + \frac{1}{2\pi}\frac{b_1^2}{2k_1k_2}\left(\frac{2\sqrt{4-\pi^2/N^2}}{6\pi^2/N^2\sqrt{\pi^2/N}}\right) \sim \frac{b_0^2 N}{2\pi^2 k_2} + \frac{b_1^2 N^3}{6\pi^4 k_1 k_2}.$$

Details of Continuance of Example 1.6.4: In this case similar computations reveals that for

$\phi(\lambda)$

$$\Gamma_N \sim \frac{1}{2k_1k_2(\alpha_1-\alpha_2)}\left(\frac{1}{\sqrt{\alpha_2(\alpha_2+4)}}-\frac{1}{\sqrt{\alpha_1(\alpha_1+4)}}\right).$$

Details of the Continuance of Example 1.7.4: In this case,

$$\begin{aligned}\Gamma_N &= \frac{1}{2\pi}\int_{\pi^2/N^2}^{4}\frac{\frac{m(m-1)}{2}k_2\lambda+k_1m}{2k_1k_2\lambda}\frac{1}{\sqrt{\lambda-\lambda^2/4}}\,d\lambda \\ &\sim \frac{m(m-1)}{8\pi k_1}\times 2\pi+\frac{m}{4k_2\pi}\times\frac{2N}{\pi}\sim\frac{m^2}{4k_1}+\frac{mN}{2k_2\pi^2}.\end{aligned}$$

Thus, based on the formula for the performance of the network of networks in (1.67), the claim is followed.

Details of Example 1.9.1: The state space matrices of the aircraft model are borrowed from [38] are given below.

$$\mathbf{A}=\begin{bmatrix}-0.003 & 0.039 & 0 & -0.322 & 0 & 0\\ -0.065 & -0.319 & 7.74 & 0 & 0 & 0\\ 0.02 & -0.101 & -0.429 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 & 0\\ 0 & -1 & 0 & 7.74 & 0 & 0\end{bmatrix}$$

$$\mathbf{B}=\begin{bmatrix}0.01 & 1\\ -0.18 & -0.04\\ -1.16 & 0.598\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix},\quad \mathbf{E}=\begin{bmatrix}0.003 & -0.039\\ 0.065 & 0.319\\ -0.02 & 0.101\\ 0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}.$$

The result of feedback gain design is

$$\mathbf{K} = \begin{bmatrix} 1.1894 & 0.7756 & -2.0834 & -7.5558 & 0.3675 & -0.2017 \\ 2.8779 & -0.0193 & 0.1032 & 0.1276 & 0.7532 & 0.0872 \end{bmatrix}.$$

For the case of observer-based relative output feedback, the following value of $\mathbf{F}$ gives us depicted performance functions.

$$\mathbf{F} = \begin{bmatrix} 9.6772 & -0.3789 \\ 1.0285 & 12.6584 \\ 0.4233 & -1.9982 \\ 0.1418 & 3.3839 \\ 9.4718 & -0.0616 \\ -0.0616 & 9.0089 \end{bmatrix}.$$

Chapter 2

Optimal Synthesis of Dynamical Networks

2.1 Introduction

In Chapter 1, we define the problem of the performance analysis of a class of multi-agent systems. Once we fully understand the relationship between the components of a dynamical network and a certain performance measure, the next question is how to effectively and in a scalable manner design or modify the network to achieve a superior performance. These design problems are the focus of this chapter.

In this context, Dai and Mesbahi have provided a classification of such design problems [46]. Moreover, Lin and co-authors have looked at the application of ADMM in design of consensus network [47]. They have also studied the optimal design of the networks based on nearest-neighbor interactions [48]. Dhingra et. al have studied finding sparse representations of consensus networks [49]. Siami and Motee have studied the sparsification of the consensus networks [50] using the randomized method of sparsification using effective resistances [51]. Several authors have also looked at the problem of growing a network for optimal performance [52–55]. The leader selection design problem also have been looked at in the literature [34, 56].

We would like to see how we can take into account the nodal dynamics and spectral

expressions for the performance. Leveraging the spectral structure of the performance measure, with respect to the spectra of the graph Laplacian is the key observation that lets us develop design tools that respect and benefit these aspects.

2.2 Problem Statement

Consider a dynamical network with agents whose dynamics are given by (1.3), but the control law instead of consensus-like law (1.4) is replaced by

$$u_i = -\mathbf{K} \sum_{j \in \mathcal{N}_i} a_{ij}(y_i - y_j) + a_{ii}\, y_i, \tag{2.1}$$

where the agents with $a_{ii} > 0$ are said to be connected to a *leader* or global source of information[1]. Moreover, we suppose that the measurement noises are zero. We can show that this control law also correspond to the form (1.4), where in this case $\mathbf{L}$ is a loopy (or grounded) Laplacian for which the self-loop weights a_{ii}'s should be accounted as well. The goal of this control law, instead of synchronization or consensus, is the stabilization of the whole network. Therefore, we define the regulated output in this case; that is

$$\nu = (\mathbf{I}_N \otimes \mathbf{C})x. \tag{2.2}$$

For an asymptotically stable network, starting from any initial condition, the state $x(t)$ vanishes as time increases, so does the regulated output ν. Again, we use $G(s)$ to denote the transfer matrix from disturbance vector ξ to this performance output ν. Then, we can still use the definition of (1.8) for the performance measure $\rho(\mathbf{L}, \mathbf{K})$ in this case. On the other hand, let us denote the transfer matrix the disturbance vector ξ to the input vector u by $F(s)$. We define the input measure to be

$$\rho_u(\mathbf{L}, \mathbf{K}) := \|F(s)\|_{\mathcal{H}_2}^2\,, \tag{2.3}$$

[1] We have removed the time argument from the variables in this chapter as well.

which reflects the magnitude of the control input that is required because of the disturbance in the network. Given these performance metrics, the following are certain generic design problems for these dynamical networks.

Problem 2.2.1 (Sparsification)**.** *Given a dynamical network over a graph $\mathcal{G}$, find a sparse graph $\mathcal{G}_{\rm s} \subset \mathcal{G}$ such that the performance of the network over the sparse graph $\mathcal{G}_{\rm s}$ is within a desired bound.*

Problem 2.2.2 (Reweighting)**.** *Given a dynamical network over a graph $\mathcal{G}$, and find the optimal weights of the graph while the sum of the weights remains constant.*

Problem 2.2.3 (Feedback Gain Design)**.** *Given a dynamical network over a fixed graph $\mathcal{G}$, find the feedback gain $\mathbf{K}$ that minimizes the objective function $J(\mathbf{K})$ that is a combination of the performance and input measures in the form of*

$$J(\mathbf{K}) := \rho(\mathbf{L}, \mathbf{K}) + \gamma \rho_u(\mathbf{L}, \mathbf{K}), \tag{2.4}$$

for some $\gamma > 0$.

We observe that the magnitude of the parameter γ represents how much we favor the magnitude of the control effort across the network.

The *research problems* are to characterize performance and input measure (1.8) in terms of grounded Laplacian eigenvalues of the underlying communication graph of the network, and to tackle these problems using the structure of these dynamical networks.

2.3 Performance Characterization

The following result is akin to Theorem 1.4.2 in the previous chapter for the case of grounded Laplacian matrices (although in simpler forms without measurement noise).

Theorem 2.3.1. *The performance measure of a asymptotically stable network without measurement noise can be expressed as*

$$\rho(\mathbf{L}, \mathbf{K}) = \sum_{i=1}^{N} \phi(\lambda, \mathbf{K})$$

for some performance function $\phi(\lambda, \mathbf{K}) = \mathrm{Tr}(\mathbf{C}\mathbf{P}(\lambda, \mathbf{K})\mathbf{C}^T$*, where* $P(\lambda, \mathbf{K})$ *is the unique positive-define solution to*

$$\mathbf{A}_\lambda \mathbf{P} + \mathbf{P}\mathbf{A}_\lambda^T + \mathbf{B}\mathbf{B}^T = 0.$$

where $\mathbf{A}_\lambda := \mathbf{A} - \lambda \mathbf{B}\mathbf{K}$*, and function* ϕ *is a rational function of* λ *and entries of* $\mathbf{K}$*.*

For the subsequent analysis, we assume the rational description of a performance function is denoted by

$$\phi(\lambda, \mathbf{K}) = \frac{\sum_{i=0}^{n_1} a_i \lambda^i}{\sum_{i=0}^{n_2} b_i \lambda^i}, \tag{2.5}$$

where a_i and b_i depend on the components of the feedback gain matrix as well.

2.4 Spectral Sparsification

Suppose that a loopy graph $\mathcal{G}$ for a multi-agent system with Laplacian matrix $\mathbf{L}$ is given. We are interested in finding a sparse approximation $\mathbf{L}_s$, where $\mathbf{L}_s$ is the Laplacian matrix of a subgraph of $\mathcal{G}$, namely $\mathcal{G}_s$ and wish to enforce matrix $\mathbf{L}_s$ to be close to $\mathbf{L}$ in some sense. There are different metrics to quantify the similarity of two graphs Laplacians. If we quantify the similarity of L and $\mathbf{L}_s$ in terms of their spectra, then the problem of finding such an approximation is called spectral sparsification [57], which has been formalized below.

Problem 2.4.1 (Spectral Sparsification)**.** *Given a loopy undirected weighted graph* $\mathcal{G}$ *with Laplacian* $\mathbf{L}$ *and* $\epsilon \in (0, 1)$*, find a sparse stabilizing subgraph* $\mathcal{G}_s \subset \mathcal{G}$ *with Laplacian* $\mathbf{L}_s$ *such that*

$$(1 - \epsilon)\, \mathbf{L} \preceq \mathbf{L}_s \preceq (1 + \epsilon)\, \mathbf{L}, \tag{2.6}$$

where $\mathbf{L}_s$ *is called an* ϵ*-sparsifier of* $\mathbf{L}$*.*

We will show this notion of sparsity is suitable to adopt for this class of networks.

2.4.1 Spectral Sparsification for Multi-Agent Systems

As discussed in the previous chapter, assume the dynamics of subsystems require a minimum connectivity threshold $\tilde{\lambda}(\mathbf{K}) \geq 0$. We are interested in design of the feedback gain $\mathbf{K}$ such that after replacing $\mathbf{L}$ with its ϵ-sparsification $\mathbf{L}_s$, not only we have a stable system, but also we are able to derive bounds on the variation in the performance measure that is

$$\Delta\rho := \rho(\mathbf{L}_s, \mathbf{K}) - \rho(\mathbf{L}, \mathbf{K}).$$

Here, we use the performance functions and find bounds on the performance measure for the network with the sparse graph.

Theorem 2.4.2. *Consider a multi-agent system with connectivity threshold $\tilde{\lambda}(\mathbf{K})$. Assume $\mathbf{L}_s$ is an ϵ-sparsifier of $\mathbf{L}$, where for some $c > 1$, the minimum eigenvalue of $\mathbf{L}$ satisfies*

$$\lambda_1(\mathbf{L}) \geq \frac{c\tilde{\lambda}(\mathbf{K})}{1-\epsilon}, \tag{2.7}$$

then the performance measure is bounded according to

$$\sum_{i=1}^{N} \beta(\lambda_i; \epsilon) \leq \Delta\rho \leq \sum_{i=1}^{N} \alpha(\lambda_i; \epsilon), \tag{2.8}$$

where the functions α and β are

$$\alpha(\lambda; \epsilon) := \max_{\hat{\lambda} \in\ \lambda \cdot [1-\epsilon, 1+\epsilon]} \phi(\hat{\lambda}) - \phi(\lambda),$$

$$\beta(\lambda; \epsilon) := \min_{\hat{\lambda} \in\ \lambda \cdot [1-\epsilon, 1+\epsilon]} \phi(\hat{\lambda}) - \phi(\lambda).$$

Moreover, we may write

$$|\Delta\rho| \leq \sum_{i=1}^{N} \eta(\lambda_i; \epsilon), \tag{2.9}$$

where the function η is given by

$$\eta(\lambda;\epsilon) := \max\{|\alpha(\lambda;\epsilon)|, |\beta(\lambda;\epsilon)|\}.$$

Furthermore, the error vanishes at $\epsilon = 0$ as $\eta(\lambda_i; 0) = 0$.

Remark 8. In some sense, c is a robustness margin that we need from the minimum connectivity threshold $\tilde{\lambda}(\mathbf{K})$. In fact, to be able to replace a dense graph by its ϵ-sparsification, one should design $\mathbf{K}$ such that (2.7) holds for the margin c.

In a special case, computing η is simplified.

Corollary 2.4.3. *In Theorem 2.4.2, if $\phi(\lambda)$ is convex and decreasing then it holds that*

$$\eta(\lambda;\epsilon) = \alpha(\lambda;\epsilon).$$

2.4.2 Spectral Sparsification by Effective Resistances

Theorem 2.4.2 endorses the suitability of the spectral sparsification for these multi-agent networks. In [51], Spielman and Srivastava have introduced a random sampling algorithm, which uses the effective resistances of the graph to produce a sparse ϵ-approximation of the graph Laplacian. In this subsection, we modify their method to allow for self-loops in the graph, while preserving the properties of the main algorithm. First, we look at a definition of effective resistance for a loopless graph Laplacian $\mathbf{L}$.

Definition 2.4.4 (Effective Resistance)**.** *The effective resistance of an edge $\{i, j\} \in \mathcal{E}$ is*

$$R_{ij} := (e_i - e_j)^T \mathbf{L}^\dagger (e_i - e_j). \tag{2.10}$$

One could start with other definitions for the effective resistance, and the Definition 2.4.4 would be a corollary (e.g. see [58]). Now, we alter it for the resistances induced by a connected loopy graph with a graph Laplacian $\mathbf{L}$.

Definition 2.4.5 (Loopy Effective Resistance)**.** *The loopy effective resistance of an edge*

Algorithm 1 Loopy Laplacian Sparsification

Input: Graph Laplacian $\mathbf{L} \succ 0$, Spectral Bound $\epsilon > 0$
Output: Sparsified Graph Laplacian $\mathbf{L}_s$
Initialize: $q = O(N \log N/\epsilon^2)$, $\mathbf{L}_s = 0$
for $i = 1$ to q **do**
Randomly Sample an edge $\{i, j\}$ with probabilities

$$p_{ij} = \frac{a_{ij} R_{ij}}{N}$$

Add the edge $\{i, j\}$ with a weight of a_{ij}/qp_{ij} to $\mathbf{L}_s$
end for

$\{i, j\} \in \mathcal{E}$ *is*

$$R_{ij} := \begin{cases} (e_i - e_j)^T \mathbf{L}^{-1} (e_i - e_j) & \text{if } i \neq j \\ e_i^T \mathbf{L}^{-1} e_i & \text{if } i = j \end{cases} \tag{2.11}$$

This is the definition that has been reported in [59] as well. The sparsification method is given by Algorithm 1, which is essentially identical to the algorithm proposed in [51] as we describe now. We start with an empty graph and for q iterations, we will sample a link with probabilities proportional to the effective resistance of that link, and add it to the graph with an adjusted weight without replacement. The difference of the algorithms would be in their definition of effective resistance.

The next theorem discusses the operational guarantee of the method, which is inherited from the original algorithm (see of [51]).

Theorem 2.4.6. *For an $\epsilon \in (1/\sqrt{N}, 1)$, with probability at least $1/2$, Algorithm 1 creates an ϵ-sparsifier $\mathbf{L}_s$ with at most $q = O(N \log N/\epsilon^2)$ links.*

Our proposed sparsification algorithm has an additional interesting attribute: Algorithm 1 identically treats the self-loops and links between distinct nodes of the graph. So it simultaneously makes them sparser such that the modified graph Laplacian is spectrally close to the original graph Laplacian with a high probability.

2.4.3 Concentration Inequality for Total Weight

We denote the total weight of the graph $\mathcal{G}$ by T, that is

$$\mathrm{T} := \sum_{\{i,j\}\in\mathcal{E}} a_{ij}.$$

We can define a similar quantity for the sparse graph $\mathcal{G}_s$, and denote it by T_s. Now, we derive an inequality, which bounds the probability that T_s deviates from T by more than a threshold δ.

Lemma 2.4.7. *The output of Algorithm 1 with q samples satisfies $\mathbb{E}\{\mathrm{T}_s\} = \mathrm{T}$ and the concentration inequality*

$$\mathbb{P}\{|\mathrm{T}_s - \mathrm{T}| > \delta\} \leq \frac{1}{q\delta^2}\left(N \sum_{(i,j)\in\mathcal{E}} \frac{a_{ij}}{R_{ij}} - \mathrm{T}^2\right).$$

We can similarly show that by replacing N in the result of Lemma 2.4.7 with $(N-1)$, we get the inequality for the original algorithm in [51] for loopless graphs using definition of the effective resistances in (2.4.4) and $p_{ij} = a_{ij}R_{ij}/(N-1)$.

2.5 Optimal Reweighting of Graph Couplings

The problem is how to redistribute the weights of a given graph, while the sum of the weights remains constant to achieve the optimal performance measure, where this constant sum of weights can be considered as a *budget constraint*. We define the design variable $W \in \mathbb{R}^{|\mathcal{E}_c|}$ for this purpose, that is the weight vector of graph of candidate links $\mathcal{G}_c \subset \mathcal{G}$ with edge set $\mathcal{E}_c \subset \mathcal{E}$; i.e.

$$W := \mathrm{vec}(\{a_{ij} : \{i,j\} \in \mathcal{E}_c\}).$$

Problem 2.5.1 (Optimal Reweighting)**.** *Given an initial loopy graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, a)$ and a*

feedback gain matrix $\mathbf{K}$ *solve the optimization problem*

$$\begin{aligned} &\underset{W}{\text{minimize}} \quad \rho(\mathbf{L},\mathbf{K}) + \gamma\rho_u(\mathbf{L},\mathbf{K}), \\ &\text{subject to: } W_j \geq 0,\ W^T\mathbf{1}_M = \mathrm{b}, \\ &\qquad W = \mathrm{vec}(\{a_{ij} : \{i,j\} \in \mathcal{E}\}) \\ &\qquad \mathbf{L} \text{ is the Laplacian of } \mathcal{G} \text{ with } \mathcal{E} \subseteq \mathcal{E}_i. \end{aligned} \tag{2.12}$$

The constraint on the sum of the weights makes it essentially a resource-allocation problem of the following form.

Problem 2.5.2 (Optimal Resource Allocation)**.** *Given an objective function* $J(x)$ *and constant budget* $b > 0$*, solve*

$$\begin{aligned} &\underset{x}{\text{minimize}} \quad J(x), \\ &\text{subject to: } x_j \geq 0,\ x^T\mathbf{1}_M = \mathrm{b}. \end{aligned} \tag{2.13}$$

2.5.1 Selective Bi-Coordinate Method

The coordinate descent methods for this problem have been recently studied; for instance see [60, 61]. Here, first we describe a method called Selective Bi-coordinate Method (SBM) [61] and then introduce an efficient way of implementation for it. It turns out that the components of this method can be customized the benefit from the structure of our objective function and its gradient (see the next subsection).

Method Summary: The optimization consists of a sequence of subprograms. Each subprogram is identified with two parameters: ϵ and δ. For the design variable x, in each step, first we should find an eligible pair of coordinates (i, j) with $i, j \in \{1, \ldots, M\}$ such that we can *transfer* the weight from the link x_i to the link x_j. These eligible candidates are characterized by the following definition.

Definition 2.5.3. *For positive parameters* $\epsilon, \delta > 0$*, an ordered pair of distinct coordinates* (i, j) *(*$i, j \in \{1, 2, \ldots, M\}$*) is called an* ϵ*-*δ *choice of coordinates at* $x \in \mathbb{R}^m$ *for function*

Algorithm 2 Selective Bi-Coordinate Method [61]

Input: Initial Point x^0, Parameters $\theta, \beta > 0$,
Sequences $\epsilon_l, \delta_l > 0$ with $\epsilon_l \downarrow 0,\ \delta_l \downarrow 0$
Output: Sequence of Design Variables $\{x_l\}$
Initialize: $x = x^0$
for $l = 1, 2, \ldots$ **do**
Set : $k = 0$, $\epsilon = \epsilon_l$, $\delta = \delta_l$
while $\exists (i, j)$, an ϵ_l-δ_l coordinate **do**
Search Direction $d_k = e_j - e_i$,
Dir. Derivative $g_{ij} = [\partial J/\partial x_j(x) - \partial J/\partial x_i(x)]$
$p = \min \left\{ p \in \mathbb{Z}_+ : J(x + \theta^p \epsilon d_k) \leq J(x) + \beta \theta^p \epsilon g_{ij} \right\}$
$x \leftarrow x + \theta^p \epsilon d_k,\ \ k \leftarrow k + 1$
end while
Subprogram output $x^l = x$
end for
return $\{x_l\}$

$J(x)$ *if*

$$x_i \geq \epsilon \text{ and } \frac{\partial J}{\partial x_i}(x) - \frac{\partial J}{\partial x_j}(x) \geq \delta.$$

We can interpret an ϵ-δ choice of coordinates as a pair of coordinates (i, j) such that component x_i of the design variable is large enough to lose at least ϵ and give it to component x_j, and also the magnitude of the directional derivative for this exchange of weight is at least δ. Once we have such a pair of coordinates, we get a search direction to conduct a backtracking line-search and find the step-size that gives us sufficient descent. We continue these weight exchanges until we run out of such eligible choices of coordinates. This will trigger the next subprogram, in which ϵ and δ are smaller (and eventually vanishing). In Algorithm 2, these steps are summarized, which comes with the following guarantee.

Theorem 2.5.4 (See [61])**.** *The sequence* $\{x^l\}$ *generated by Algorithm 2 has stationary limit points. Moreover, if* $J(x)$ *is convex,*

$$\lim_{l \to \infty} J(x^l) = J^*,$$

where J^* *is the optimal value of Problem 2.5.2.*

2.5.2 Efficient Implementation by Structure Exploitation

To use the Algorithm 2 for reweighting the networks, we need efficient ways of finding ϵ-δ pairs of coordinates (i.e. two eligible links to do the weight exchange), the value of the performance measure in the line-search routine and the directional gradients.

Solution: We can address these tasks by keeping track of certain auxiliary matrices during the optimization procedure, which we will identify shortly. The basic idea that makes this possible is the specific structure of the updates to Laplacian during this optimization and how these updates are propagated into the performance measure and its gradient with respect to the design variable.

Consider the exchange of the weight of amount Δw from the link i to link j according to the described coordinate descent method. It can be decomposed into two rank-one updates to the Laplacian in the form of

$$\mathbf{L}_{\text{new}} = \mathbf{L}_{\text{old}} - \Delta w \cdot \mathbf{e}_i \mathbf{e}_i^T + \Delta w \cdot \mathbf{e}_j \mathbf{e}_j^T,$$

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are the vectors corresponding to those links in the graph incidence matrix. Now, we analyze the propagation of such rank-one update into the performance measure and its gradient.

Theorem 2.5.5. *Assume we update the graph Laplacian* $\mathbf{L}$ *with a rank-one matrix* $\mathbf{R}$*. Then, the performance measure of the dynamical network evolves according to*

$$\begin{aligned}&\rho(\mathbf{L}+\mathbf{R},\mathbf{K}) = \rho(\mathbf{L},\mathbf{K}) + \\ &\operatorname{Tr}\left(\mathcal{R}_1(\mathbf{L})\sum_{k=1}^{2^{n_2}} Z_k + \sum_{k=1}^{2^{n_1}} W_k \mathcal{R}_2^{-1}(\mathbf{L}) + \sum_{k=1}^{2^{n_1}} W_k \sum_{k=1}^{2^{n_2}} Z_k\right),\end{aligned}$$

where the involved matrices are defined using the coefficients of the performance function in (2.5) and are given by

$$\mathcal{R}_1(\mathbf{L}) = \sum_{i=0}^{n_1} a_i \mathbf{L}^i,\ \mathcal{R}_2(\mathbf{L}) = \sum_{i=0}^{n_2} b_i \mathbf{L}^i$$

which possess the direct and inverse updates

$$\mathcal{R}_1(\mathbf{L}+\mathbf{R}) = \mathcal{R}_1(\mathbf{L}) + \sum_{k=1}^{2^{n_1}} \mathbf{W}_k,$$
$$\mathcal{R}_2^{-1}(\mathbf{L}+\mathbf{R}) = \mathcal{R}_2^{-1}(\mathbf{L}) + \sum_{k=1}^{2^{n_2}} \mathbf{Z}_k,$$

for rank-one matrices $\mathbf{W}_k$ *and* $\mathbf{Z}_k$*. Moreover, the update for* $\mathcal{R}_2^{-1}(\mathbf{L})$ *could be found by consecutive applications of the Sherman-Morrison formula.*

Thus, if we keep track of $\mathcal{R}_1(\mathbf{L})$ and $\mathcal{R}_2^{-1}(\mathbf{L})$ and how they are updated according to the theorem, we can find the value of the updates to the performance measure during the line search procedure in Algorithm 2.

Now, note that the gradient of the performance measure with respect to its design variable W can be evaluated as

$$g := \nabla_W \rho(\mathbf{L}, \mathbf{K}) = \operatorname{diag}\left(\mathbf{B}_c \nabla_{\mathbf{L}} \rho(\mathbf{L}, \mathbf{K}) \mathbf{B}_c^T\right), \tag{2.14}$$

where $\mathbf{B}_c$ is the incidence matrix of the candidate sets. In the following lemma, we see that the intermediate term, Jacobian with respect to $\mathbf{L}$, can be characterized via the derivative of the performance function and Laplacian spectra.

Lemma 2.5.6. *We can compute the gradient of the performance measure with respect to the Laplacian matrix* $\mathbf{L}$ *is*

$$\nabla_{\mathbf{L}} \rho(\mathbf{L}, \mathbf{K}) = \mathbf{U} \operatorname{diag}\left(\frac{d\phi}{d\lambda}(\lambda_i)\right) \mathbf{U}^T.$$

Remark 9. Lemma 2.9.1 asserts that the gradient of the performance measure with respect to the graph Laplacian, which is a symmetric matrix has essentially the same eigenvectors as $\mathbf{L}$, while while eigenvalues are the values of $d\phi/d\lambda$ at the Laplacian eigenvalues. We may deduce the same formula for the input measure ρ_u; i.e.

$$\nabla_{\mathbf{L}} \rho_u(\mathbf{L}, \mathbf{K}) = \mathbf{U} \operatorname{diag}\left(\frac{d\mu}{d\lambda}(\lambda_i)\right) \mathbf{U}^T.$$

Now, we see a similar result for the update of $\nabla_{\mathbf{L}}\rho(\mathbf{L},\mathbf{K})$, which can be carried into g using equation (2.14).

Theorem 2.5.7. *Assume we express the rational function $d\phi/d\lambda$ with similar representation for $\phi(\lambda)$ in Theorem 2.5.5. Then, after a rank-one update $\mathbf{R}$ to the Laplacian, the Jacobian of the performance measure with respect to $\mathbf{L}$ evolves according to*

$$\nabla_{\mathbf{L}}\rho(\mathbf{L}+\mathbf{R},\mathbf{K}) = \nabla_{\mathbf{L}}\rho(\mathbf{L},\mathbf{K}) + \mathcal{R}_1(\mathbf{L})\sum_{k=1}^{2^{n_2}}\mathbf{Z}_k + \sum_{k=1}^{2^{n_1}}\mathbf{W}_k\mathcal{R}_2^{-1}(\mathbf{L}) + \sum_{k=1}^{2^{n_1}}\mathbf{W}_k\sum_{k=1}^{2^{n_2}}\mathbf{Z}_k,$$

where the matrices are defined similar to Theorem 2.5.5.

If we keep track of $\mathcal{R}_1(\mathbf{L})$ and $\mathcal{R}_2^{-1}(\mathbf{L})$ for $\nabla_{\mathbf{L}}\rho(\mathbf{L}+\mathbf{R},\mathbf{K})$ and their updates according to this theorem, then we can find the updated value of g after each step. This also facilitates finding coordinates that are eligible for weight exchange.

Lemma 2.5.8. *Assume we have the objective function gradient g according to (2.14). Set i and j to be*

$$i = \operatorname*{argmax}_k \{g_k : \ W_k \geq \epsilon\}.$$

$$j = \operatorname*{argmin}_k \{g_k\}.$$

If $g_i - g_j \geq \delta$, then (i,j) is an ϵ-δ pair of coordinates. Otherwise, there is no such pair of coordinates.

Hence, we can use Lemma 2.5.8 to find out when the next subprogram is supposed to start (i.e. when we have no more eligible ϵ-δ pairs of coordinates).

2.5.3 Running Time Analysis

The updates to matrices $\mathcal{R}_1$ and $\mathcal{R}_2^{-1}$ based on the performance function $\phi(\lambda)$ and its derivative $d\phi(\lambda)/d\lambda$ require $O(N^2)$. Because the matrix products in the updates for the performance measure and its gradient in Theorem 2.5.5 and 2.5.7 deal with rank-one matrices, they can be done in $O(M)$. Then, we need to a sort the links find the corresponding to maximum and minimum directional derivatives, which can be done in $O(M \log M)$. We

can keep track of the links with weights more than ϵ at no additional cost. Hence, each step would require $O(M \log M + N^2)$. The initial computation of the matrix $\mathcal{R}_2^{-1}$ would require $O(N^3)$ operations.

2.6 Network Growth

Adding k number of links from a given set of candidate links $\mathcal{E}_c$, with $|\mathcal{E}_c| = M_{\mathbf{L}}$ to the graph of a multi-agent system is an interesting combinatorial problem where we have

$$\binom{M_{\mathbf{L}}}{k}$$

choices, where the number of possible choices grows exponentially as k starts increasing from 1. One may formalize this problem as follows.

Problem 2.6.1 (Optimal Network Growth)**.** *Given a loopy graph $\mathcal{G}$ with Laplacian $\mathbf{L}$, and a set of candidate links represented via $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M\}$, find the additive unweighted graph $\mathcal{G}_a$ with edge set $\mathcal{E}_a$ and Laplacian matrix $\mathbf{L}_a$ from the optimization problem*

$$\begin{aligned} &\underset{\mathcal{G}_a}{\textit{minimize}} \quad \rho(\mathbf{L} + \mathbf{L}_a, \mathbf{K}) \\ &\textit{subject to: } |\mathcal{E}_a| = k \end{aligned} \tag{2.15}$$

We seek tractable and scalable approximations to this problem. In this section, we show that the structure of the problem lets us efficiently implement the greedy algorithm.

2.6.1 Greedy Algorithm with Efficient Implementation

The greedy algorithm for adding k links to the graph of the network is well-known. We add k links one by one, while at each step, we look for the link with maximum improvement in the performance measure. If we see no improvement within the remaining set of candidates, we terminate the algorithm. In Algorithm 3, we have summarized these steps.

We can exploit the structure of rank-one updates in this case as well: we keep track of matrices $\mathcal{R}_1(\mathbf{L})$ and $\mathcal{R}_2^{-1}(\mathbf{L})$ for the performance measure according to Theorem 2.5.5.

Algorithm 3 Greedy Network Growth

Input: Laplacian $\mathbf{L}$, Set of Candidates $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M\}$
Output: Modified Laplacian $\mathbf{L}$
  Set : $S = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_M\}$
  for $i = 1, 2, \ldots k$ **do**
    Find index j_s

$$\mathbf{e}_{j_s} = \underset{\mathbf{e}_j \in S}{\operatorname{argmin}} \ \rho(\mathbf{L} + \mathbf{e}_j \mathbf{e}_j^T, \mathbf{K})$$

    if $\rho(\mathbf{L} + \mathbf{e}_j \mathbf{e}_j^T, \mathbf{K}) \geq \rho(\mathbf{L}, \mathbf{K})$ **then**
      return L
    end if

    Update the Laplacian $\mathbf{L} \leftarrow \mathbf{L} + \mathbf{e}_{j_s} \mathbf{e}_{j_s}^T$,
    Remove it from the candidates $S \leftarrow S \backslash \{\mathbf{e}_{j_s}\}$
  end for

  return L

Then, for all candidate links, finding the change in the performance measure when adding the link can be done according to the updates in Theorem 2.5.5. Thus, we do not need to compute a full set of eigenvalues of $\mathbf{L}$ for each candidate link.

Remark 10. We do not impose any assumption on the form of the performance function (e.g. monotonicity or convexity).

2.6.2 Running Time Analysis

Computing the change in the value of performance measure can be done in $O(N)$ because the matrices involved are of rank one. Doing this for all candidate links implies we need at most $O(MN)$ operations. Next, we need the updates for matrices $\mathcal{R}_1$ and $\mathcal{R}_2^{-1}$ that similar to previous requires $O(N^2)$ operations. Sorting the changes in the performance measure after the update requires $O(M \log M)$. Hence, adding each link requires $O(N^2 + MN + M \log M) = O(MN)$ operations. The initialization evaluation of $\mathcal{R}_2^{-1}$ would require $O(N^3)$ operations.

2.7 Optimal Feedback Gains Design

The optimal feedback gains design problem for a given topology is is formalized below.

Problem 2.7.1 (Optimal Feedback Gains Design)**.** *Given a loopy graph Laplacian* $\mathbf{L}$*, find a feedback gain* $\mathbf{K}$ *solving*

$$\underset{\mathbf{K}}{\textit{minimize}} \quad \rho(\mathbf{L},\mathbf{K}) + \gamma\rho_u(\mathbf{L},\mathbf{K}) \tag{2.16}$$

$$\textit{subject to: } \tilde{\lambda}(\mathbf{K}) < \lambda_1 \tag{2.17}$$

$\mathbf{K}$ *induces an unbounded stability region.*

The design variable of this problem is $\mathbf{K} \in \mathbb{R}^{p\times n}$, thus independent of number of subsystems, the search space is often small. Thus, without customized algorithms, the general nonlinear optimization software (e.g. trust-region or interior-point methods) may effectively address this problem, where the constraints are different based on application of Routh-Hurwitz criteria for the characteristic polynomial of the subsystems

$$\dot{x} = (\mathbf{A} - \lambda\mathbf{B}\mathbf{K})x.$$

Because the pair $(\mathbf{A},\mathbf{B})$ is stabilizable, the Routh-Hurwitz criteria must gives us necessary and sufficient conditions for existence of minimum connectivity threshold on $\mathbf{K}$ in form of a set of nonlinear inequalities on elements of $\mathbf{K}$ (see the examples of previous chapter).
Gradient and Hessian Evaluation: Assume K_{jk} and K_{lo} represent the entries of $\mathbf{K}$ with $i,l \in \{1,\dots,p\}$ and $j,o \in \{1,\dots,n\}$. The partial derivatives of first and second-order of $\rho(\mathbf{L},\mathbf{K})$ with respect to elements of $\mathbf{K}$

$$\frac{\partial(\rho+\gamma\rho_u)}{\partial K_{jk}} = \sum_{i=1}^{N} \frac{\partial(\phi+\gamma\mu)}{\partial K_{jk}}, \tag{2.18}$$

$$\frac{\partial^2(\rho+\gamma\rho_u)}{\partial K_{jk}\partial K_{lo}} = \sum_{i=1}^{N} \frac{\partial^2(\phi+\gamma\mu)}{\partial K_{jk}\partial K_{lo}}(\lambda_i,\mathbf{K}), \tag{2.19}$$

To numerically evaluate the gradient and Hessian of the objective function with respect to $\mathbf{K}$, we need to find these function of $\mathbf{K}$ and λ, an then substitute for the given value of Laplacian eigenvalues.

2.8 Numerical Studies

Finally, we look at numerical implementations of different schemes that have been investigated throughout this Chapter.

2.8.1 Examples on Spectral Sparsification

In the first example, we derive the parametric bounds for spectral sparsification for certain subsystems.

Example 2.8.1. Consider a network of single-integrators (see Example 1.6.1 in the previous chapter). We find the functions $\alpha(\lambda;\epsilon)$ and $\beta(\lambda;\epsilon)$ to bound variation of the performance measure of the system upon sparsification. Following the corresponding definitions

$$\alpha(\lambda;\epsilon) = \frac{\epsilon}{2k(1-\epsilon)\lambda},\ \beta(\lambda;\epsilon) = \frac{-\epsilon}{2k(1+\epsilon)\lambda},$$

we can evaluate functions

$$\eta(\lambda;\epsilon) = \max(|\alpha(\lambda;\epsilon)|, |\beta(\lambda;\epsilon)|) = \frac{\epsilon}{2k(1-\epsilon)\lambda}.$$

Note that we may find the last using Lemma 2.4.3 as well. For this specific case, it turns out that the relative absolute error can be bounded solely in terms of ϵ, because

$$\frac{|\Delta\rho|}{\rho(\mathbf{L},\mathbf{K})} \leq \frac{\sum_{i=1}^{N}\eta(\lambda;\epsilon)}{\sum_{i=1}^{N}\phi(\lambda,\mathbf{K})} = \frac{\epsilon}{1-\epsilon}. \tag{2.20}$$

For double integrators, those functions are

$$\alpha(\lambda;\epsilon) = \frac{\epsilon(2-\epsilon)}{2k_1k_2(1-\epsilon)^2\lambda^2}, \beta(\lambda;\epsilon) = \frac{-\epsilon(2+\epsilon)}{2k_1k_2(1+\epsilon)^2\lambda^2},$$
$$\eta(\lambda;\epsilon) = \frac{\epsilon(2-\epsilon)}{2k_1k_2(1-\epsilon)^2\lambda^2},$$

where the last one is true by application of Lemma 2.4.3. A similar result in this case also

	$\mathbf{L}$	$\mathbf{L}_s$	U.B.	L.B.
$n=1$	3.3203	3.2516	2.0752	8.3007
$n=2$	10.2864	8.7953	4.0181	64.2901
$n=3$	3.4037	2.8021	1.1201	84.9530

Table 2.1: Performance measure before and after sparsification with nodal dynamics chosen from single to triple integrators.

hold, that is

$$\frac{|\Delta\rho|}{\rho(\mathbf{L},\mathbf{K})} \leq \frac{\epsilon(2-\epsilon)}{(1-\epsilon)^2}. \tag{2.21}$$

We may repeat this procedure for triple-integrators (see Example 1.6.3), where because the performance function in this case is convex and decreasing in $(\tilde{\lambda},\infty)$, computing the bound functions for spectral sparsification is similar to previous cases.

In the next example, we analyze the implementation of Algorithm 1 for finding sparse approximations to a loopy graph Laplacian.

Example 2.8.2. We start from a dense interconnection graph, which is formed by joining the following undirected random graphs:
a) a random graph with $N = 80$ nodes where the nodes are distributed randomly in the space and are connected to their neighbors if their distance is less than $r = 0.3$.
b) a graph of self-loops where each self-loops exist with independent probability of 0.3.

The sample graph that we work with has 1667 links between distinct nodes and 18 self-loops. We look at a sample result of Algorithm 6 applied on the Laplacian matrix of this graph with $\epsilon = 0.6$. The graph before and after the sparsification has been illustrated in Fig. 2.1, where number of links between distinct nodes and self-loops decrease to 723 and 10, respectively. Moreover, In Fig. 2.2, we have shown the ratio of the eigenvalues of the sparsified Laplacian $\mathbf{L}_s$ to the initial Laplacian matrix $\mathbf{L}$, which confirms

$$\frac{\lambda_i(\mathbf{L}_s)}{\lambda_i(\mathbf{L})} \in [1-\epsilon, 1+\epsilon].$$

Now we look at the changes in the performance of networks of single to triple integrators when replacing the $\mathcal{G}$ with its ϵ-sparsifie $\mathcal{G}_s$. The feedback gain design, for single and double

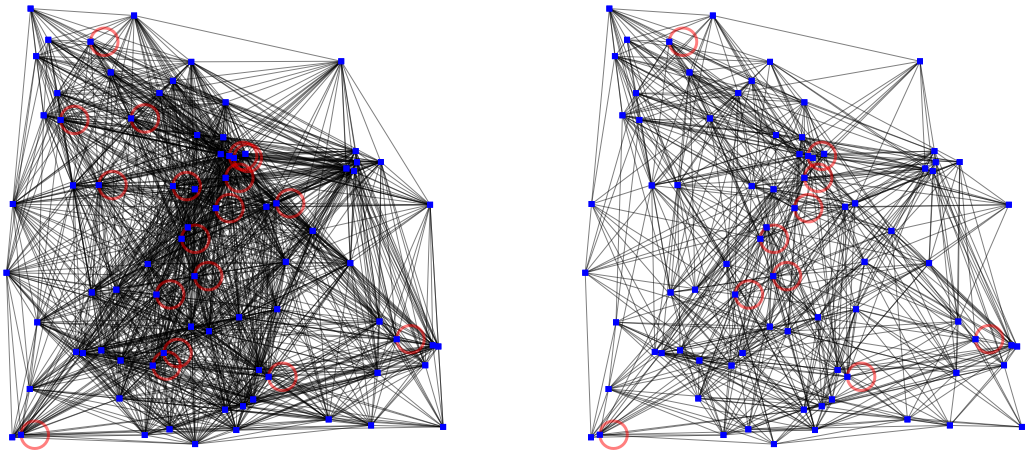

Figure 2.1: This plot shows the sparsity pattern of the original graph versus the sparsified graph (the self-loops are more visible in the colored version).

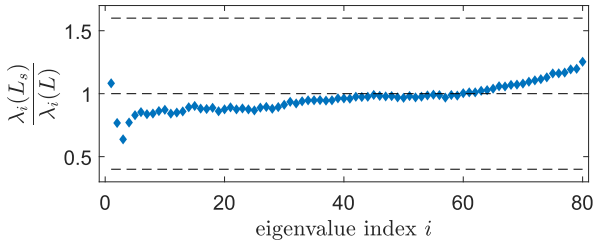

Figure 2.2: The ratio of eigenvalues of the sparsified Laplacian to the eigenvalues of original Laplacian. The dashed the top and bottom dashed lines depict the expected bounds of this ratio, $1 + \epsilon$ and $1 - \epsilon$.

	$\mathbf{L}$	$\mathbf{L}_s$
Weight of Loop Links	18	20.5122
Weight of Internal Links	1667	1653.4

Table 2.2: The weights of the links before and after sparsification.

integrators is straight-forward, thus we set all the gains components equal to 1. For the network of triple-integrators, we design $\mathbf{K} = [k_1, k_2, k_3]$ such that after sparsification, the system remains asymptotically stable. Based on the requirements, we let $k_1 = 1$ and $k_3 = 3$. Because $\tilde{\lambda}(\mathbf{K}) = k_1/k_2k_3$, based on Theorem 2.4.2, we set $c = 1.2$ and

$$k_2 = \frac{ck_1}{k_3\lambda_1(1-\epsilon)} \approx 4.5283.$$

The results together with bounds on the performance has been shown in Table 2.1, where the bounds are not tight, while the performance of the system in each case is close to the original system.

We can compare the total weight of the graph $\mathcal{G}$ with $\mathcal{G}_s$, that has been shown in Table 2.2 that are close (total difference about 11.1). In fact, using Theorem 2.4.7 we can write

$$\mathbb{P}\{|\mathrm{T}_s - \mathrm{T}| > \delta\} \leq \frac{1}{q\delta^2}\left(N\sum_{(i,j)\in\mathcal{E}}\frac{a_{ij}}{R_{ij}} - \mathrm{T}^2\right) \approx \frac{149.21}{\delta^2}.$$

For instance, this implies that for $\delta \approx 17.3$, the bound becomes $\mathbb{P}\{|\mathrm{T}_s - \mathrm{T}| > 17.3\} \leq 1/2$.

In the next example, we show when the derived bounds can be helpful in practice.

Example 2.8.3. Reconsider the graph that was sparsified in Example 2.8.2. We rerun Algorithm 1 on $\mathcal{G}$ for $200,000$ different samples and look at the performance measure of the multi-agent systems with different integrator dynamics over the $\mathcal{G}_s$. The empirical distributions of the performance measure have been illustrated in Fig. 2.3. We observe that lower and upper-bounds on the performance correspond to *worst-case* bound of multi-agent system upon sparsification and the probability that the outcome of this algorithm induces a performance measure close to the worst-case is small.

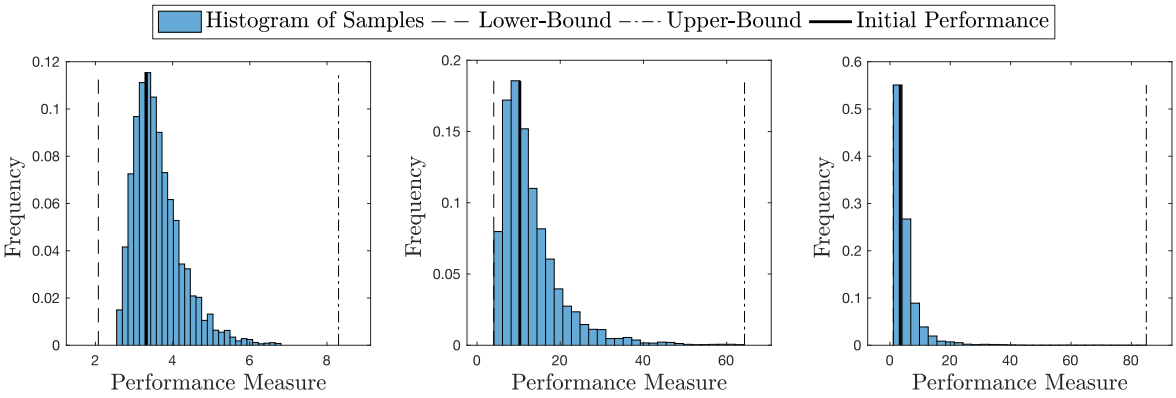

Figure 2.3: Experimental distribution of the performance measure of the multi-agent system with sparisified graph Laplacian produced in Example 2.8.3, for single, double, and triple integrators from left to right.

2.8.2 Examples on Feedback Gain Design and Reweighting

Example 2.8.4. We reconsider the settings of Example 1.6.5 for a platoon of third-order vehicles. Recall that the performance function in this case is given by

$$\phi(\lambda) = \frac{1}{2k_1} \frac{k_3\lambda + 1}{k_2k_3\lambda^3 + (k_2 - k_1\tau)\lambda^2}.$$

We apply Theorem 2.5.5 and write

$$\mathcal{R}_1(\mathbf{L}) = k_3\mathbf{L} + \mathbf{I}_N \Rightarrow \mathcal{R}_1(\mathbf{L} + R) = \mathcal{R}_1(\mathbf{L}) + k_3\mathbf{R}.$$

and also

$$\begin{aligned}
&\mathcal{R}_2(\mathbf{L}) = 2k_1\left(k_2k_3L^3 + (k_2 - k_1\tau)\mathbf{L}^2\right) \Rightarrow \\
&\mathcal{R}_2(\mathbf{L} + \mathbf{R}) = \mathcal{R}_2(\mathbf{L}) + 2k_1(k_2 - k_1\tau)(\mathbf{LR} + \mathbf{RL} + \mathbf{R}^2) \\
&+ 2k_1k_2k_3(\mathbf{L}^2\mathbf{R} + \mathbf{LRL} + \mathbf{LR}^2 + \mathbf{RL}^2 + \mathbf{RLR} + \mathbf{R}^2\mathbf{L} + \mathbf{R}^3).
\end{aligned}$$

Now, by consecutive application of the Sherman-Morrison updates, once we keep track of $\mathcal{R}_1(\mathbf{L})$, $\mathcal{R}_2(\mathbf{L})$, and $\mathcal{R}_2^{-1}(\mathbf{L})$ we can compute the updates to the performance measure according to the theorem. In fact, in this case, we have 10 rank-one updates for $R_2^{-1}(\mathbf{L})$.

Now, for the gradient computations, we can compute that

$$\frac{d\phi(\lambda)}{d\lambda} = \frac{-1}{2k_1k_2}\left\{\frac{k_3^3\,(1-a)}{a^2\,(k_3\lambda+a)^2} + \frac{k_3\,(1-a)}{a^2\lambda^2} + \frac{2}{a\lambda^3}\right\}.$$

where $a = \dfrac{k_2 - k_1\tau}{k_2}$. This partial fraction has been done to make the analysis more tractable by denoting R_{i1} and R_{i2} for the term i. For the first term, we can define

$$\begin{aligned}
&\mathcal{R}_{11}(\mathbf{L}) = \frac{-k_3^3(1-a)}{2a^2k_1k_2}\mathbf{I}_N \Rightarrow \mathcal{R}_{11}(\mathbf{L}+\mathbf{R}) = \mathcal{R}_{11}(\mathbf{L}),\\
&\mathcal{R}_{12}(\mathbf{L}) = k_3^2\mathbf{L}^2 + 2ak_3\mathbf{L} + a^2\mathbf{I}_N \Rightarrow\\
&R_{12}(\mathbf{L}+\mathbf{R}) = \mathcal{R}_{12}(\mathbf{L}) + k_3^2\left(\mathbf{LR}+\mathbf{RL}+\mathbf{R}^2\right) + 2ak_3\mathbf{R}.
\end{aligned}$$

The second term can be similarly treated using

$$\begin{aligned}
&\mathcal{R}_{21}(\mathbf{L}) = \frac{-k_3(1-a)}{2a^2k_1k_2}\mathbf{I}_N \Rightarrow \mathcal{R}_{21}(\mathbf{L}+\mathbf{R}) = \mathcal{R}_{21}(\mathbf{L}),\\
&\mathcal{R}_{22}(\mathbf{L}) = \mathbf{L}^2 \Rightarrow R_{22}(\mathbf{L}+\mathbf{R}) = \mathcal{R}_{22}(\mathbf{L}) + \mathbf{LR} + \mathbf{RL} + \mathbf{R}^2.
\end{aligned}$$

And finally for the third term

$$\begin{aligned}
&\mathcal{R}_{31}(\mathbf{L}) = \frac{-1}{ak_1k_2}\mathbf{I}_N \Rightarrow \mathcal{R}_{31}(\mathbf{L}+\mathbf{R}) = \mathcal{R}_{31}(\mathbf{L}),\\
&\mathcal{R}_{32}(\mathbf{L}) = \mathbf{L}^3 \Rightarrow R_{32}(\mathbf{L}+\mathbf{R}) = \mathcal{R}_{32}(\mathbf{L}) + \mathbf{L}^2\mathbf{R} + \mathbf{LRL} + \mathbf{LR}^2 + \mathbf{RL}^2 + \mathbf{RLR} + \mathbf{R}^2\mathbf{L} + \mathbf{R}^3.
\end{aligned}$$

Thus, for each rank one update to $\mathbf{L}$, we have to do a rank 13 update to the Jacobian with respect to $\mathbf{L}$

In the next example, we consider numerical experiments testing the feedback gain design and reweighting algorithms.

Example 2.8.5. Again, we consider the platoon of vehicles described in Example 1.6.5. We set the response constant as $\tau = 1/2$ and as an initial guess for the gains we choose $\mathbf{K} = [1, 2, 1]$, which has an unbounded stability region and $\tilde{\lambda}(\mathbf{K}) = 0$. For this feedback

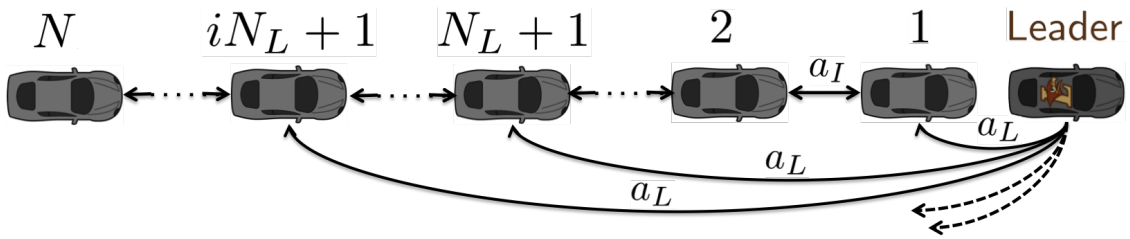

Figure 2.4: An example for the interconnection graph of a platoons

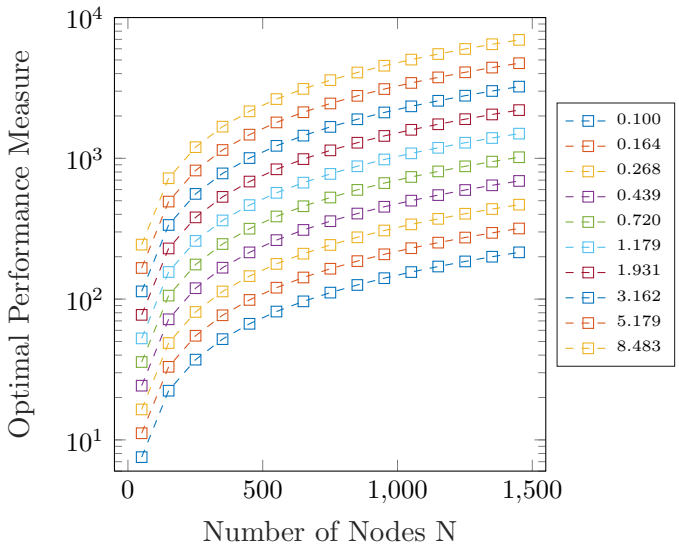

Figure 2.5: The optimal values of performance measure for different values of N and γ.

gain, the resulting performance function is given by

$$\phi(\lambda) = \frac{\lambda + 1}{\lambda^2(4\lambda + 3)},$$

that is convex and decreasing for $\lambda > \tilde{\lambda}(\mathbf{K}) = 0$

Initial Graph: We can describe the topology on which we will apply the reweighting and feedback gain design as follows: a string of N cars with a weighted graph shown in Fig. 2.4. Each car is connected to two vehicles in the front and back (expect for the first and last cars) with weight a_I. Moreover, periodically, the vehicles number $iN_L + 1$, $i = 0, 1, \ldots$ are connected to the leader with weight a_L.

Feedback Gain Design: The nonlinear constraint (1.104) defines a part of boundary of stability for the feedback gain design problem that is necessary and sufficient to hold for

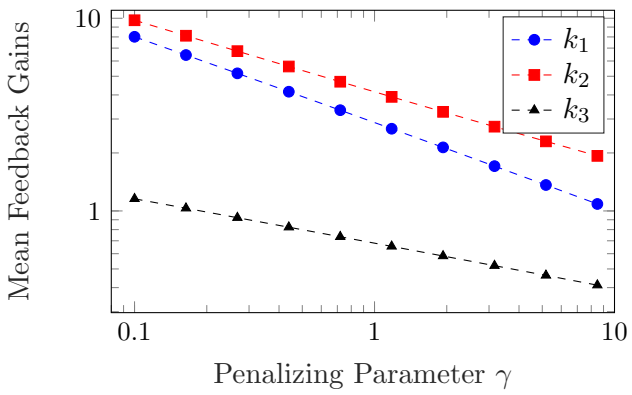

Figure 2.6: The mean of optimal values of the entries of feedback gain K for the platooning dynamics. The variance (for different values of N) is relatively negligible in this scale, so we have omitted error-bars on the data points.

λ_1, hence we write it as

$$c(\mathbf{K}) := \tau k_1 - k_3 k_2 \lambda_1 + k_2 \leq 0.$$

For a chosen initial topology with the Laplacian $\mathbf{L}$, the feedback gain design problem becomes

$$\underset{\mathbf{K}}{\text{minimize}} \quad \sum_{i=1}^{N} \frac{1}{2k_1} \frac{k_3\lambda_i + 1}{k_2 k_3 \lambda_i^3 + (k_2 - k_1\tau)\lambda_i^2} + \gamma \sum_{i=1}^{N} \frac{\tau k_2^2 \lambda_i + k_2 k_3^2 \lambda_i^2 - k_1 \tau k_3 \lambda_i + k_1 \tau}{2\tau(k_2 - k_1\tau + k_2 k_3 \lambda_i)}. \tag{2.22}$$

subject to: $k_1, k_2 > 0,\ k_3 \geq 0,$

$$c(\mathbf{K}) = \tau k_1 - k_3 k_2 \lambda_1 + k_2 < 0. \tag{2.23}$$

It is trivial to observe that computing the gradient of the objective function an the constraints in terms of the $\mathbf{K}$ will only involves simple differentiation. We implement a trust-region method with `fmincon` function of MATLAB. The start point is $\mathbf{K} = [1, 2, 1]$. We set the spacing between the agents $N_L = 10$, the weight of the links between the agents $a_I = 1$ and the weight of the self-loops $a_L = 3$. We evaluate the solutions for $N \in \{51, 101, \ldots, 1451\}$. We show the optimal value of the performance measure (i.e. not the objective function that involves a term due to input penalization) in Fig. 2.5 for different values of γ and N.

We observe that for this class of topologies, for a fixed value of γ, the optimal feedback

gain was relatively insensitive to the number of nodes (not all values listed in this chapter). In fact, the maximum variance of the optimal values of k_i's for each value of γ for different N's was about 10^{-3}. Hence, it Fig 2.6, for different value of γ, we have demonstrated the trend of mean optimal value of the feedback gains, where the limit of large γ corresponds to the case where we are interested in minimal energy to stabilize the multi-agent system.

Topology Reweighting: We freeze the links from the leader to the grounded vehicles and reweight the links between the cars with an initial value of $a_I = 1$ for the case we have no penalization due to input; i.e. for $\gamma = 0$.

The optimization parameters for implementation of Algorithm 2 at subprogram number $l = 1, 2, \dots$ are

$$\delta_l = 1/l,\ \epsilon_l = 1/l,\ \beta = \theta = 0.5.$$

The initial topology has the spacing between the grounded nodes $N_L = 5$, the internodal links have the weight $a_I = 1$, while the grounded agents have the self-loops of weight $a_L = 3$. We run the optimization problem for different network sizes, which are chosen as

$$N \in \{351, 501, \dots, 1551\},$$

and we let the optimization to continue for 4 subprograms of each $400(1+\log \frac{N}{351})$ iterations. The initial and final value of $\rho(\mathbf{L}, \mathbf{K})$ for each platoon size N are shown in Fig. 2.7.

For $N = 351$, a sample of optimization progress for 4 subprograms of 400 iterations is shown in Fig. 2.8. We may look at the distribution of the degrees of the graph between the agents (without the portion of the degree due to connection to the leader) after the optimization. A sample solution is depicted in Fig. 2.9, revealing that the reweighting favors adding the weight to agents that are grounded (i.e. connected to the leader). Then, it favors the rest of the agents based on their proximity to the grounded vehicles. The optimal allocation of the weights is almost periodic (between the grounded vehicles) and symmetric. In Fig. 2.10, we have illustrated the weights of the optimal links between the nodes, where the grounded nodes are demonstrated with black color. In fact, we have identified three kind of links in this graph, where the strong links are connected to the grounded links.

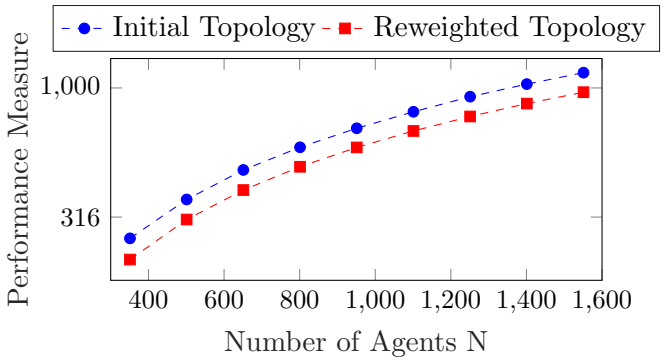

Figure 2.7: The performance measure before and after reweighting

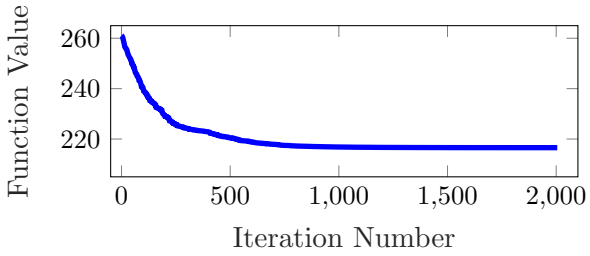

Figure 2.8: The performance measure versus number of iterations.

Then weaker links, but not the weakest, are connected to the nodes that are connected to the grounded medium with one intermediate agent. And finally, the weakest links that are the farthest from the grounded vehicles.

2.9 Conclusion and Discussion

Appendix: Proofs

Proof of 2.4.2. First we show that the growth of the performance function ϕ is limited by

$$\beta(\lambda_i;\epsilon) \leq \phi(\lambda_i(\mathbf{L}_s)) - \phi(\lambda_i) \leq \alpha(\lambda_i;\epsilon), \tag{2.24}$$

If this holds, then we can write

$$|\phi(\lambda_i(\mathbf{L}_s)) - \phi(\lambda_i)| \leq \eta(\lambda_i;\epsilon), \tag{2.25}$$

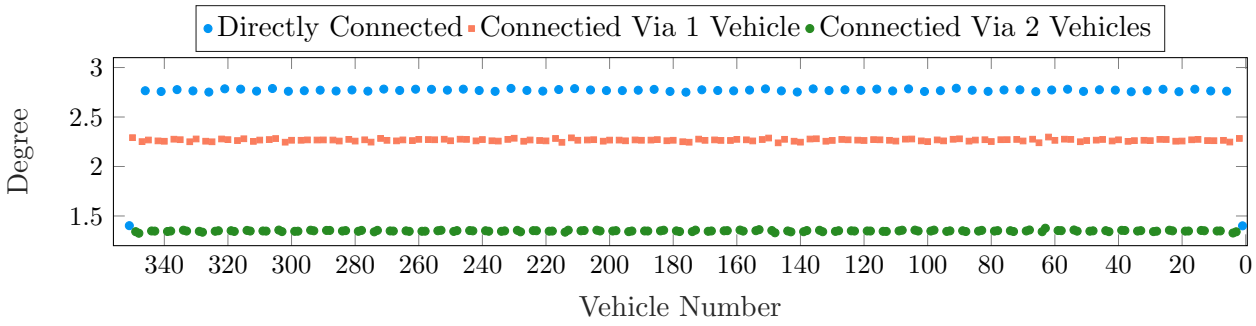

Figure 2.9: A sample representation of the degrees of the graph between the agents after the reweighting procedure.

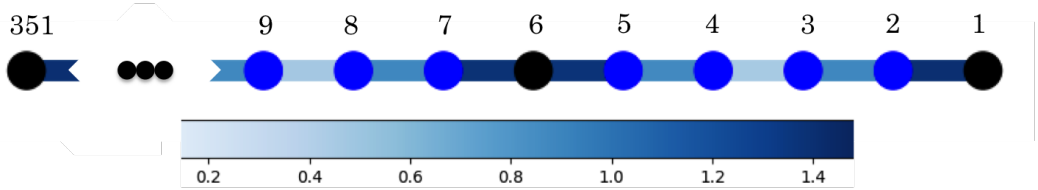

Figure 2.10: An illustration from the optimal weights of the links in the case of $N = 351$. The grounded nodes are denoted by black color, and are spaced by 5 vehicles.

If $\mathbf{L}_s$ is an ϵ-sparsifier of matrix L, then

$$\lambda_i(\mathbf{L}_s) \in \lambda_i \times [1-\epsilon, 1+\epsilon],$$

that is a compact interval, so we can look at the maximum and minimum of ϕ to get corresponding point-wise inequalities. Now, we may use the terms in the inequalities above to prove the claimed inequalities of the theorem. For the second part, based on the definition of α and β, once ϵ is set to zero, the interval $\lambda_i \times [1-\epsilon, 1+\epsilon]$ shrinks to the singleton λ_i, so the deviation of function ϕ from $\phi(\lambda_i)$ is zero. The fact that η vanishes as well is immediate from its definition. □

Proof of Theorem 2.4.6. The steps needed for proof of this theorem are exactly the same steps provided in the proof of the algorithm for loopless graph Laplacians in [51] that was rephrased in [57] (see the Proof of Theorem 5 therein). Indeed, almost all of the steps in that proof are true if we replace $\mathbf{L}^\dagger$ (in the case of loopless Laplacian) with $\mathbf{L}^{-1}$ (in the case of Loopy Laplacian), so we do not repeat it here. In addition to extending the definition of effective resistances to allow for self-edges, the other change is that for an undirected loopy

Laplacian, it holds that

$$\sum_{\{i,j\}\in\mathcal{E}} a_{ij}R_{ij} = N, \tag{2.26}$$

while this sum for a loopless Laplacian matrix we have [51]

$$\sum_{\{i,j\}\in\mathcal{E}} a_{ij}R_{ij} = N - 1. \tag{2.27}$$

Now, we prove (2.26), where its prove is also similar to proof of (2.27) in [51], where we define the projection matrix as

$$\mathbf{\Pi} := \mathbf{W}^{1/2}\mathcal{B}\mathbf{L}^{-1}\mathcal{B}^T\mathbf{W}^{1/2}.$$

The interesting fact about this matrix is that for the edge number k (based on numbering in the incidence matrix $\mathcal{B}$) between nodes i and j, we have

$$\Pi_{kk} = w_{ij}R_{ij},$$

which implies we may write

$$\sum_{k=1,\ldots,|\mathcal{E}|} \Pi_{kk} = \mathrm{Tr}(\mathbf{\Pi}) = \sum_{\{i,j\}\in\mathcal{E}} a_{ij}R_{ij}.$$

It is easy to find that $\mathbf{\Pi}^2 = \mathbf{\Pi}$, so its eigenvalues are either 1 or 0. If the graph is connected, the rank of $\mathbf{\Pi}$ would be N. Hence, it has exactly N eigenvalues equal to 1, and the rest of them are zero. Hence, $\mathrm{Tr}(\mathbf{\Pi}) = N$, and (2.26) follows. This justifies the difference of the probabilities used in the original algorithm from those that we use in Algorithm 6, where they used to be

$$p_{ij} = \frac{a_{ij}R_{ij}}{N},$$

while in the modified algorithms we should set

$$p_{ij} = \frac{a_{ij} R_{ij}}{N-1}.$$

□

Proof of Lemma 2.4.7. We may define a sequence of i.i.d. positive random variables $\{z_i\}_{i=1,2,\ldots}$, where each random variable is identical to $z : \Omega \to \mathbb{R}_{++}$, with a probability mass function $\pi : \mathbb{R}_{++} \to \mathbb{R}_+$ such that

$$\pi\left(\frac{a_{ij}}{q p_{ij}}\right) = p_{ij}.$$

In words, the random variable z describes the random weight that is added to the graph at each sampling. We can compute the expectation of this random variable

$$\mu := \mathbb{E}\{z\} = \sum_{(i,j)\in\mathcal{E}} p_{ij} \frac{a_{ij}}{q p_{ij}} = \frac{1}{q} \sum_{(i,j)\in\mathcal{E}} a_{ij} = \frac{1}{q}\mathrm{T}.$$

We can compute its variance as

$$\sigma^2 := \mathrm{Var}(z) = \mathbb{E}\{z^2\} - \mathbb{E}\{z\}^2 = \sum_{(i,j)\in\mathcal{E}} p_{ij} \frac{a_{ij}^2}{q^2 p_{ij}^2} - \frac{\mathrm{T}^2}{q^2}$$

$$\sum_{(i,j)\in\mathcal{E}} \frac{N a_{ij}^2}{q^2 a_{ij} R_{ij}} - \frac{\mathrm{T}^2}{q^2} = \frac{1}{q^2}\left(N \sum_{(i,j)\in\mathcal{E}} \frac{a_{ij}}{R_{ij}} - \mathrm{T}^2\right).$$

Now, we define the sum

$$S_q := \sum_{i=1}^{q} z_i = \mathrm{T}_s,$$

where $\mathbb{E}\{S_q/q\} = \mathbb{W}/q$ and $\mathrm{Var}(S_q/q) = \sigma^2/q$. Applying the Chebyshev's inequality to the random variable S_q/q, we may write

$$\mathbb{P}\left\{\left|\frac{S_q}{q} - \mathbb{E}\left\{\frac{S_q}{q}\right\}\right| > \frac{\delta}{q}\right\} \leq \frac{\sigma^2/q}{(\delta/q)^2},$$

which is equivalent to

$$\mathbb{P}\left\{|S_q - \mathbb{E}\{S_q\}| > \delta\right\} = \mathbb{P}\left\{|\mathbb{W}_s - \mathbb{W}| > \delta\right\} \leq \frac{\sigma^2 q}{\delta^2}.$$

Substituting the value of σ^2 in the above inequality proves the claim of the theorem. □

Proof of Lemma 2.5.6. The author of [62] defines the following two notations:

- a region Ω in $\mathbb{R}^N$ is symmetric if for any permutation matrix $\mathbf{P} \in \mathbb{R}^{N\times N}$, we have $\mathbf{P}\Omega = \Omega$ (i.e. closed under permutations).
- a function $f : \Omega \to \mathbb{R}$ a symmetric function if for any $x \in \Omega$ and permutation matrix $\mathbf{P}$, it holds that

$$f(\mathbf{P}x) = f(x).$$

Now, we may mention the result of [62].

Proposition 2.9.1 ([62])**.** *Let $\Omega \in \mathbb{R}^n$ be open and symmetric, and suppose function $f(x) : \Omega \to \mathbb{R}$ is symmetric. Then the spectral function $f(\lambda(.))$ is differentiable at the symmetric matrix $\mathbf{X}$ if and only if function f is differentiable at the vector $\lambda(\mathbf{X})$. If so, the gradient of $f \circ \lambda$ at the matrix $\mathbf{X} = \mathbf{X}^*$ is*

$$\nabla_X f(\mathbf{X}^*) = \mathbf{U}\text{diag}\left(\nabla_x f(\lambda(\mathbf{X}^*))\right)\mathbf{U}^T,$$

where the matrix X^* can be decomposed using a unitary matrix $\mathbf{U}$ as $X^* = \mathbf{U}\lambda(\mathbf{X})\mathbf{U}^T$.

We apply this for computation of the gradient of the performance measure with respect to the graph Laplacian. Note that $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, and that we may define function $f(x)$ that we need to use according to Proposition 2.9.1 is

$$\rho(\mathbf{L}, \mathbf{K}) = \sum_{i=1}^{N} \phi(\lambda_N) := f(\lambda(\mathbf{L})),$$

where one can note that

$$\nabla_x f(\lambda(\mathbf{L})) = \left(\frac{\mathrm{d}\phi}{\mathrm{d}\lambda}(\lambda_1), \ldots, \frac{\mathrm{d}\phi}{\mathrm{d}\lambda}(\lambda_N)\right)^T,$$

which proves the claim upon application of Lemma 2.9.1. □

Proof of Theorem 2.5.5. We may write

$$\rho = \mathrm{Tr}\left(\left(\sum_{i=0}^{N_1} a_i \mathbf{L}^i\right)\left(\sum_{i=0}^{N_2} b_i \mathbf{L}^i\right)^{-1}\right) := \mathrm{Tr}(\mathcal{R}(\mathbf{L})).$$

Now, we focus on the argument of the trace. The rank one update to $\mathbf{L}$, namely $\mathbf{R}$ spreads into the trace argument as

$$\mathcal{R}(\mathbf{L}+\mathbf{R}) := \mathcal{R}_1(\mathbf{L}+\mathbf{R})\mathcal{R}_2^{-1}(\mathbf{L}+\mathbf{R}) = \left(\sum_{i=0}^{N_1} a_i(\mathbf{L}+\mathbf{R})^i\right)\left(\sum_{i=0}^{N_2} b_i(\mathbf{L}+\mathbf{R}^T)^i\right)^{-1}.$$

Now, one can inspect that expansion of the binomial series gives

$$\mathcal{R}(\mathbf{L}+\mathbf{R}) = \left(\sum_{i=0}^{N_1} a_i\mathbf{L}^i + \sum_{i=1}^{N_1}\sum_{j=1}^{2^i-1} a_i\mathbf{V}_{ij}\right)\left(\sum_{i=0}^{N_2} b_i\mathbf{L}^i + \sum_{i=1}^{N_2}\sum_{j=1}^{2^i-1} b_i\mathbf{V}_{ij}\right)^{-1},$$

where $\mathbf{V}_{ij}$ describe all the rank one updates that will appear in the expansion of $(\mathbf{L}+uv^T)^i$, which are 2^i-1 different rank one matrices in general. We can arbitrarily order the rank one terms that are added and write a more simple form

$$\mathcal{R}(\mathbf{L}+\mathbf{R}) = \left(\sum_{i=0}^{N_1} a_i\mathbf{L}^i + \sum_{k=1}^{2^{N_1}} \mathbf{W}_k\right)\left(\sum_{i=0}^{N_2} b_i\mathbf{L}^i + \sum_{k=1}^{2^{N_2}} \mathbf{W}_k\right)^{-1}.$$

Now, we can apply 2^{N_2} times the Sherman-Morrison update formula for the matrix $\mathcal{R}_2(\mathbf{L})$, which results in

$$\mathcal{R}(\mathbf{L}+\mathbf{R}) = \left(\mathcal{R}_1(\mathbf{L}) + \sum_{k=1}^{2^{N_1}} \mathbf{W}_k\right)\left(\mathcal{R}_2^{-1}(\mathbf{L}) + \sum_{k=1}^{2^{N_2}} \mathbf{Z}_k\right) =$$
$$\mathcal{R}(\mathbf{L}) + \mathcal{R}_1(\mathbf{L})\sum_{k=1}^{2^{N_2}} \mathbf{Z}_k + \sum_{k=1}^{2^{N_1}} \mathbf{W}_k\mathcal{R}_2^{-1}(\mathbf{L}) + \sum_{k=1}^{2^{N_1}} \mathbf{W}_k \sum_{k=1}^{2^{N_2}} \mathbf{Z}_k.$$

Now, we can find the update to the performance measure that is

$$\rho(\mathbf{L}+\mathbf{R},\mathbf{K}) = \rho(\mathbf{L},\mathbf{K}) + \operatorname{Tr}\left(\mathcal{R}_1(\mathbf{L})\sum_{k=1}^{2^{N_2}}\mathbf{Z}_k + \sum_{k=1}^{2^{N_1}}\mathbf{W}_k\mathcal{R}_2^{-1}(\mathbf{L}) + \sum_{k=1}^{2^{N_1}}\mathbf{W}_k\sum_{k=1}^{2^{N_2}}\mathbf{Z}_k\right).$$

□

Proof of Theorem 2.5.7. Using Lemma 2.5.6, one could equivalently write the gradient as

$$\nabla_{\mathbf{L}}\rho(\mathbf{L}+\mathbf{R},\mathbf{K}) = \left(\sum_{i=0}^{N_1} a_i\mathbf{L}^i\right)\left(\sum_{i=0}^{N_2} b_i\mathbf{L}^i\right)^{-1}.$$

The rest the proof is similar to the proof of Theorem 2.5.5. □

Chapter 3

Randomly Switching Consensus Networks

3.1 Introduction

In the last two chapters, we have modelled the interaction between the agents to be perfect, which happen over a fixed graph. However, in practice there is always randomness associated with the interactions between the subsystems. For instance, the random nature of the wireless communication links could be a source of randomness in the topology of the network [63]. Therefore, in applications such as mobile robotics, we should be concerned about the implications of the randomness that could be echoed in the performance of the dynamical network.

One of the first papers which considered the consensus problem over random graphs is [15], where the authors study the stability and convergence rate of a class of these networks. The stability criteria of the random networks over ergodic graph processes were later further characterized; for instance see [16, 64]. More recently, the convergence speed of a these networks has been again brought to the attention [65, 66]. The stability of the consensus algorithm over Markov chain of graphs has been also studied in the literature [67].

In this chapter, we consider the distributed consensus or synchronization algorithms for simple integrator agents with random interactions between the agents. We assume that these

interconnections according to a random process that rollouts in an identically distributed manner (in time). Unlike the settings of the previous chapters, we consider the problem in discrete-time. We evaluate a performance measure for the quality of synchronization. It turns out that evaluating the performance measure suffers from undesired computational complexity. Therefore, we find a tight lower-bound on the performance measure that can be evaluated in comparatively less time. Moreover, we layout the problem of *optimal random switching*, which focuses on the best way to switch between a number of graph topologies in random in term of the performance.

3.2 Notations

The main body of notations are the ones that are provided in Section 1.2, which are completed for this chapter. The set of doubly-stochastic matrices of appropriate dimension are denoted by $\mathcal{S}$. The Frobenius norm of a matrix is denoted by $\|.\|_F$. The maximum eigenvalue of matrix $\mathbf{A}$ is denoted by $\lambda_{\max}(\mathbf{A})$.

3.3 Problem Statement

Let us consider a network with scalar discrete-time integrators agents, which means that the state of agent i at time $t \in \mathbb{Z}_+$ is denoted by $x_i \in \mathbb{R}$ and it follows the dynamics

$$x_i(t+1) = x_i(t) + u_i(t) + \xi_i(t), \tag{3.1}$$

where $u_i(t) \in \mathbb{R}$ is the control input and $\xi_i(t)$ is the exogenous disturbance at time t. We collect N instances of these agents. Now, let us define the state vector of the network $x \in \mathbb{R}^N$ to be

$$x := [x_1, x_2, \dots, x_N]^T, \tag{3.2}$$

and similarly the vector of disturbances by $\xi \in \mathbb{R}^N$. Now, set some $\epsilon > 0$ and consider the control law over a sequence of undirected weighted graphs $\{\mathcal{G}_t\}_{t\in\mathbb{Z}_+}$, according to

$$u_i(t) = \epsilon \sum_{j\in\mathcal{N}_i(t)} a_{ij}(t)(x_j - x_i), \tag{3.3}$$

where $\mathcal{N}_i(t)$ denotes the set of neighbors of node i at time $t \in \mathbb{Z}_+$; or equivalently set of neighbors of node i in graph $\mathcal{G}_t$. Suppose that $\mathbf{L}(t)$ is the graph Laplacian corresponding to graph $\mathcal{G}_t$. In order to rewrite the dynamics of the network in closed-form, we use the Perron matrix corresponding to $\mathbf{L}(t)$ that is given by

$$\mathbf{W}(t) := \mathbf{I}_N - \epsilon\,\mathbf{L}(t), \tag{3.4}$$

(e.g. see [14,68]). For small enough values of ϵ, these Perron matrices are doubly-stochastic: they have nonnegative entries with rows and columns that add-up to one. We denote all these matrices by $\mathcal{S}$. Then, the dynamics of these networks are given by

$$\begin{cases} x(t+1) = \mathbf{W}(t)x(t) + \xi(t), \\ y(t) = \mathbf{C}x(t), \end{cases} \tag{3.5}$$

where $y \in \mathbb{R}^N$ is the output of the network and output matrix $\mathbf{C}$ satisfies

$$\mathbf{C}1_N = 0_N.$$

Now, we state the assumptions that characterize the random networks of interest. The first assumption characterizes the random switches that happen between finite number of topologies.

Assumption 3.3.1. *The process $\{\mathbf{W}(t)\}_{t\in\mathbb{Z}_+}$ is a random process of i.i.d. random variables, with $\mathbf{W}_0 = \mathcal{W}$, where $\mathcal{W}$ is a random variable of Perron matrices with finite range*

$$R := \{\mathbf{W}_1, \ldots, \mathbf{W}_m\} = \{\mathbf{I}_N - \epsilon\mathbf{L}_1, \ldots, \mathbf{I}_N - \epsilon\mathbf{L}_m\} \tag{3.6}$$

and probability mass function $\pi : R \to \mathbb{R}_+$.

This assumption suggests that the value of $\pi_i := \pi(\mathbf{W}_i)$ is equal to the probability that at any instant $t \in \mathbb{R}_+$, Perron matrix process $\mathbf{W}(t)$ is at realization $\mathbf{W}_i$. We can express this probability distribution using a probability vector $\Pi \in \mathbb{R}_+^m$

$$\Pi := [\pi_1, \ldots, \pi_m]. \tag{3.7}$$

For instance, let us suppose that we have $N = 5$ subsystems, where they communicate over $m = 2$ possible topologies as depicted in Fig. 3.1[1]. After we build $\mathbf{L}_1$ and $\mathbf{L}_2$, we choose ϵ and we get the values for the corresponding Perron matrices $\mathbf{W}_1$ and $\mathbf{W}_2$.

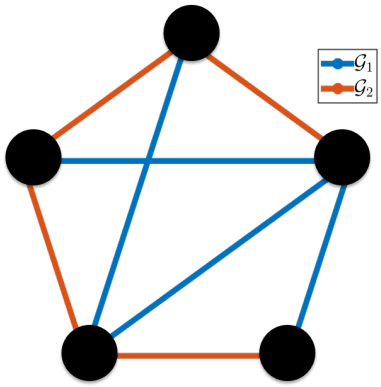

Figure 3.1: A schematic of two topologies for a random network.

The next assumption describes the properties of the additive disturbance (or noise) process ξ.

Assumption 3.3.2. *The noise is a random process* $\{\xi(t)\}_{t\in\mathbb{Z}_+}$ *that consists of i.i.d. random variables with* $\xi_0 \sim \mathcal{N}(0, \mathbf{I}_N)$.

Consider the output signal $\{y(t)\}_{t\in\mathbb{Z}_+}$ of the network that is a random process. We define the performance measure of this class of random networks to be the steady-state variance of the output, which is given by

$$\rho(\mathcal{W}) := \lim_{t\to\infty} \mathbb{E}\left\{y(t)^T y(t)\right\}, \tag{3.8}$$

[1]While there are no mutual links between two graphs in this example, in general, we do not require this to hold.

where the expectation is taken over all admissible realizations of the noise process and switches under the described settings. The amount of fluctuations of the state of the network around the consensus state is reflected in this quantity. To have an idea of what these fluctuations could look like, samples of the output of the network $y(t)$ with and without random switches based on the topologies shown in Fig. 3.1 are illustrated in Fig. 3.2.

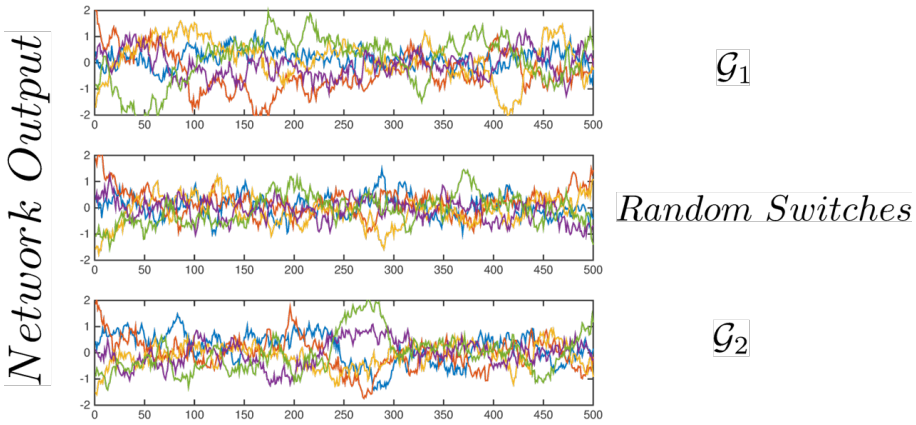

Figure 3.2: The sample output of the noisy networks with fixed and random graphs.

We observe that in this case, the switching between two topologies have enhanced the performance measure of the network (i.e., decreased the intensity of the fluctuations).

The *research problems* are to characterize the stability and performance of these random networks, find scalable bounds for the approximation of the performance measure, and characterize the optimal random switching between a number of topologies.

3.4 Performance of LTI Consensus Networks

For a linear time-invariant (LTI) network, there is no randomness and we have $\mathbf{W}(t) \equiv \mathbf{W} \in \mathcal{S}$. Then, the dynamics (3.5) become

$$\begin{cases} x(t+1) = \mathbf{W}(t)x(t) + \xi(t), \\ y(t) = \mathbf{C}x(t), \end{cases} \tag{3.9}$$

Built upon the results in [69], one can compute the performance measure of (3.9), wherein we have used the notation $\mathbf{L} := \mathbf{C}^2$.

Theorem 3.4.1. *If $\lambda_{\max}(\mathbf{M}_N\mathbf{W}) < 1$, the performance measure of an LTI consensus network (3.9) cane be expressed as*

$$\rho(\mathbf{W}) = \mathrm{Tr}\left(\mathbf{L}\left(\mathbf{I}_n - \mathbf{M}_N\mathbf{W}^2\right)^{-1}\right). \tag{3.10}$$

Moreover, if $\mathbf{C} = \mathbf{L} = \mathbf{M}_N$,

$$\rho(\mathbf{W}) = \mathrm{Tr}\left((\mathbf{I}_N - \mathbf{M}_N\mathbf{W}^2)^{-1}\right) - 1, \tag{3.11}$$

or equivalently in terms of eigenvalues of $\mathbf{W}$

$$\rho(\mathbf{W}) = \sum_{i=2}^{N} \frac{1}{1-\lambda_i(\mathbf{W})^2}, \tag{3.12}$$

where $\lambda_1 = 1$ corresponding to eigenvector 1_N is excluded from these summations.

Note that the latter special case is reported in [69]. An analogous analysis for randomly switching networks can be followed. The authors of [66] have demonstrated that

$$\lim_{t\to\infty} \mathbb{E}\left\{y_t^T y_t\right\} = \mathrm{vec}(\mathbf{M}_N)^T(\mathbf{I}_{N^2} - \mathbf{G}_{\mathcal{W}})^{-1}\mathrm{vec}(\mathbf{M}_N),$$

where $\mathbf{C} \in \mathbb{R}^{N\times(N-1)}$, $\mathbf{C}^T\mathbf{1}_N = 0$, $\mathbf{C}^T\mathbf{C} = \mathbf{I}_{n-1}$, $\mathbf{C}\mathbf{C}^T = \mathbf{M}_N$, and

$$\mathbf{G}_{\mathcal{W}} := (\mathbf{M}_N \otimes \mathbf{M}_N)\,\mathbb{E}\left\{\mathcal{W}\otimes\mathcal{W}\right\}.$$

Following a similar approach, we evaluate (3.8), where $\mathbf{K} := \mathbf{C}\otimes\mathbf{C}$.

Theorem 3.4.2. *If $\lambda_{\max}(\mathbf{G}_{\mathcal{W}}) < 1$, then the performance measure of the random network defined by (4.18) can be expressed as*

$$\rho\left(\mathcal{W}\right) = \mathrm{vec}(\mathbf{M}_N)^T\mathcal{Q}\ \mathrm{vec}(\mathbf{M}_N), \tag{3.13}$$

where $\mathcal{Q} := \mathbf{K}(\mathbf{I}_{N^2} - \mathbf{G}_{\mathcal{W}})^{-1}$.

Alternatively, we may write the expression (3.13) as

$$\rho\left(\mathcal{W}\right) = \mathrm{vec}(\mathbf{L})^T(\mathbf{I}_{N^2} - \mathbf{G}_{\mathcal{W}})^{-1}\mathrm{vec}(\mathbf{M}_N), \tag{3.14}$$

that reproduces the result of [66] if we set $\mathbf{L} = \mathbf{M}_N$.

3.5 Lower-Bound on Performance Measure

As number of the subsystems N increases, matrix $\mathbf{G}_{\mathcal{W}}$, that is a square matrix of dimension N^2 becomes huge. Consequently, it is vital to have computationally cheaper methods to (at least approximately) evaluate the performance measure. In this section, we introduce a lower-bound on the performance measure using the notion of the *Nearest Kronecker Product* (NKP) to a matrix. We show that this bound is cheaper to compute than the performance measure. First, let us formally define a nearest Kronecker product to a matrix.

Definition 3.5.1 (see [70])**.** *Let* $\mathbf{A}$*,* $\mathbf{B}^*$*, and* $\mathbf{C}^*$ *be matrices such that* $\mathbf{A}$ *and* $\mathbf{B}^* \otimes \mathbf{C}^*$ *have the same dimension. Then a Nearest Kronecker Product to* $\mathbf{A}$ *is* $\mathbf{B} \otimes \mathbf{C}$ *computed by*

$$\{\mathbf{B}, \mathbf{C}\} = \underset{\mathbf{B}^*, \mathbf{C}^*}{\mathrm{argmin}} \left\|\mathbf{A} - \mathbf{B}^* \otimes \mathbf{C}^*\right\|_F. \tag{3.15}$$

The solution of optimization problem (3.15) has interesting couplings with the properties of $\mathbf{A}$. We list a number of them that are useful for our problem.

Proposition 3.5.2 (see [70])**.** *The following statements hold true concerning the solutions to the minimization problem (3.15).*

1. *If* $\mathbf{A}$ *can be expressed as sum of Kronecker products*

$$\mathbf{A} = \sum_{i=1}^{r} \mathcal{B}_i \otimes \mathcal{C}_i,$$

   then the solutions $\mathbf{B}$ *and* $\mathbf{C}$ *are linear combinations of* $\mathcal{B}_i$*'s and* $\mathcal{C}_i$*'s, respectively.*

2. *If* $\mathbf{A}$ *is positive-semidefinite and symmetric, then we can always choose* $\mathbf{B}$ *and* $\mathbf{C}$ *to be positive-semidefinite and symmetric. If so, then it holds that* $\mathbf{A} \succeq \mathbf{B} \otimes \mathbf{C}$*.*

Next, we use this notion and propose a lower-bound on the performance of a randomly switching network. To this end, we define

$$\mathbf{E}_{\mathcal{W}} := \mathbb{E}_{\mathcal{W}} \{\mathbf{W} \otimes \mathbf{W}\}. \tag{3.16}$$

We may expand this matrix as

$$\mathbf{E}_{\mathcal{W}} = \sum_{i=1}^{m} \sqrt{\pi_i}\, \mathbf{W}_i \otimes \sqrt{\pi_i}\, \mathbf{W}_i := \sum_{i=1}^{m} \mathcal{B}_i \otimes \mathcal{B}_i. \tag{3.17}$$

Theorem 3.5.3 (Performance Lower-Bound)**.** *Suppose that symmetric $N \times N$ matrix $\hat{\mathbf{W}} \succ 0$ is given by solving the nearest Kronecker product problem given by*

$$\hat{\mathbf{W}} = \underset{\mathbf{W}}{\operatorname{argmin}} \left\| \mathbb{E} \{\mathcal{W} \otimes \mathcal{W}\} - \mathbf{W} \otimes \mathbf{W} \right\|_F. \tag{3.18}$$

If $\lambda_{\max}(\mathbf{G}_{\mathcal{W}}) < 1$, then for matrix

$$\mathbf{W}_s := \mathbf{M}_N \hat{\mathbf{W}} + \mathbf{J}_N / N \in \mathcal{S}, \tag{3.19}$$

it holds that $\lambda_{\max}(\mathbf{M}_N \mathbf{W}_s) < 1$. Moreover, we can bound the performance measure by

$$\rho(\mathbf{W}_s) \leq \rho(\mathcal{W}). \tag{3.20}$$

The significance of (3.20) is that evaluating the lower-bound requires computation of performance of an LTI network, which is cheaper for large networks (see the end of this section for a time-complexity analysis).

Labeling the eigenvalues such that $\lambda_1(\mathbf{W}_s)$ and $\lambda_1(\hat{\mathbf{W}})$ share the eigenvector $v_1 = \mathbf{1}_n$, then for $i \neq 1$, $\lambda_i(\hat{\mathbf{W}}) = \lambda_i(\mathbf{W}_s)$ and $v_i(\mathbf{W}_s) = v_i(\hat{\mathbf{W}})$. Thus, if $\mathbf{C} = \mathbf{M}_N$, (3.12) implies

$$\rho(\mathcal{W}) \geq \sum_{i=2}^{N} \frac{1}{1 - \lambda_i(\mathbf{W}_s)^2} \tag{3.21}$$

3.5.1 Computation of Lower-Bound

In the previous result, we did not sketch the numerical algorithm that lets us find the nearest Kronecker product. Here, we recall results that let us do so. For a matrix $\mathbf{A} = \sum_{i=1}^{r} \mathcal{B}_i \otimes \mathcal{C}_i$, the reshaping matrix is

$$\tilde{\mathbf{A}} := \sum_{i=1}^{r} \operatorname{vec}(\mathcal{B}_i) \operatorname{vec}(\mathcal{C}_i)^T, \tag{3.22}$$

(see [70]). Using the singular value decomposition (SVD) of $\tilde{\mathbf{A}}$, the following theorem asserts the solution of (3.15).

Theorem 3.5.4 (see [70]). *Let σ_1 be the largest singular value of $\tilde{\mathbf{A}}$ with singular vectors u_1 and v_1. Then $\mathbf{B} \in \mathbb{R}^{m\times n}$ and $\mathbf{C} \in \mathbb{R}^{p\times q}$ defined by $\operatorname{vec}(\mathbf{B}) := \sqrt{\sigma_1}u_1$ and $\operatorname{vec}(\mathbf{C}) := \sqrt{\sigma_1}v_1$ minimize the objective function $\|\mathbf{A} - \mathbf{B} \otimes \mathbf{C}\|_F$.*

Now we conduct an error analysis for this lower-bound. The error of NKP is known to be $\|\mathbf{A} - \mathbf{B} \otimes \mathbf{C}\|_F = \|\tilde{\mathbf{A}} - \operatorname{vec}(\mathbf{B})\operatorname{vec}(\mathbf{C})^T\|_F$, (see [70]), where based on the rest of singular values of $\tilde{\mathbf{A}}$,

$$\left\|\tilde{\mathbf{A}} - \operatorname{vec}(\mathbf{B})\operatorname{vec}(\mathbf{C})^T\right\|_F \leq \sqrt{\sigma_2^2 + \cdots + \sigma_r^2} := c_r,$$

and the rank of $\tilde{\mathbf{A}}$ is r. Nonzero singular values of $\tilde{\mathbf{A}}$ for $\mathbf{A} = \mathbf{E}_{\mathcal{W}}$ are at most m, since based on (3.17), $\tilde{\mathbf{A}}$ is summation of m rank-one matrices. So, we define $c_m := \sqrt{\sigma_2^2 + \cdots + \sigma_m^2}$, where $\|\mathbf{E}_{\mathcal{W}} - \hat{\mathbf{W}} \otimes \hat{\mathbf{W}}\|_F \leq c_m$. We denote the perturbations

$$\Delta\rho := \rho(\mathcal{W}) - \rho(\mathbf{W}_s),$$

$$\Delta\mathbf{G} := \mathbf{G}_{\mathcal{W}} - \mathbf{M}_N\mathbf{W}_s \otimes \mathbf{M}_N\mathbf{W}_s,$$

and also

$$\Delta^{-1}\mathbf{G} := (\mathbf{I}_{N^2} - \mathbf{G}_{\mathcal{W}})^{-1} - (\mathbf{I}_{N^2} - \mathbf{M}_N\hat{\mathbf{W}} \otimes \mathbf{M}_N\hat{\mathbf{W}})^{-1}.$$

We may show that the proposed lower-bound satisfies the following first order lower-bound.

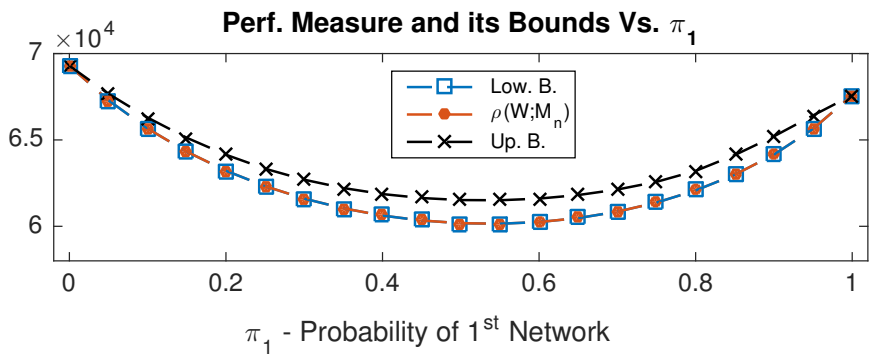

Figure 3.3: The performance measure and its bounds for Example 3.5.6.

Theorem 3.5.5 (First Order Error)**.** *For sufficiently small values of* $\|\Delta\mathbf{G}\|_2$*, it holds that*

$$\Delta\rho \leq \frac{c_m \|\mathrm{vec}(\mathbf{M}_N)\mathrm{vec}(\mathbf{L})^T\|_2}{(1-\lambda_{\max}^2(\mathbf{M}_N\mathbf{W}_s))^2}. \tag{3.23}$$

If $\mathbf{C} = \mathbf{L} = \mathbf{M}_N$, then we get

$$\Delta\rho \leq \frac{c_m(N-1)}{(1-\lambda_{\max}^2(\mathbf{M}_N\mathbf{W}_s))^2}. \tag{3.24}$$

3.5.2 Examples of the Lower-Bound Computation

We conclude this section with several different examples, where we will apply Theorem 3.5.4 on $\mathbf{A} = \mathbf{E}_{\mathcal{W}}$ to compute the lower-bounds. The first example considers a case when the lower-bound is tight, but the error analysis is not.

Example 3.5.6 (Lower and Upper Bound). Consider a random network with two possible topologies. Therefore, the set of Perron matrices is given by

$$R = \{\mathbf{W}_1, \mathbf{W}_2\},$$

with $\Pi = [\pi_1, \pi_2]$ denoting the probabilities that each topology occurs. We choose $N = 100$ and compute the Perron matrices as follows. We set

$$\mathbf{W}_i = \mathbf{I}_n - \epsilon\mathbf{L}_i,$$

wherein $\epsilon = 10^{-4}$) and $\mathbf{L}_i$'s correspond to Erdos-Renyi random graphs with $|\mathcal{E}_1| = 466$ and $|\mathcal{E}_2| = 438$ edges. Knowledge of π_1 fixes the performance measure, since $\pi_1 + \pi_2 = 1$. For $\pi_1 \in [0, 1]$, we compute the performance measure, its lower-bound, and upper-bound using (3.24). The results are shown in Fig. 3.3. The maximum relative error of the lower-bound is less than $10^{-9}\%$ (i.e. practically zero), while the upper-bound has a maximum relative error of 22.6%. The lower-bound and upper-bound were calculated in about 2.5 seconds seconds, while the performance measure was evaluated in about 105.2 seconds.

Remark 11. Example 3.5.6 motivates the design of a randomly switching protocol instead of sticking to a single topology. Indeed, as shown in Fig. 3.3, by switching, for intermediate values of π_1, we outperform each network on their own. This inspires us to formulate the optimal random switching problem (see Section 3.6).

Example 3.5.7 (Random Switches between Three Networks). We rework Example 3.5.6 for three networks. Here, we have a probability vector $\Pi = [\pi_1, \pi_2, \pi_3]$ with shown in Fig. 3.4, $N = 40$, $|\mathcal{E}_1| = 103$, $|\mathcal{E}_2| = 109$, and $|\mathcal{E}_3| = 114$. The Perron matrices are

$$\mathbf{W}_i = \mathbf{I}_N - 1/(2N-2)\mathbf{L}_i$$

. We compute the performance measure and lower-bound for $\pi_1, \pi_2 \in [0, 1]$, which are represented in Fig. 3.5, with a maximum relative error less than 0.51%.

For large scale networks, the evaluation of (3.13) might become computationally tedious. However, Example 3.5.8 suggests that the lower-bound might still be cheap to compute.

Example 3.5.8 (Large Networks). Consider a random switching among the doubly stochastic matrices $R = \{\mathbf{W}_1, \mathbf{W}_2\}$ with equal probability; i.e. $\Pi = [1/2, 1/2]$. For a range of N, we create two Erdos-Renyi graphs with edge probability $p = \log(N)/N$ and the Perron matrices with $\epsilon = 1/(2N-2)$. We evaluate the performance measure by Monte-Carlo simulations, using the sample trajectories until $t_{\max} = 3000$ for 300 different samples. On the other hand, we compute the lower-bound for each N. The performance measures from simulations, lower-bounds, and their machine times have been illustrated in Fig. 3.6. The maximum relative error of the lower-bound with respect to the simulations value was less than 0.56% in this sample experiment.

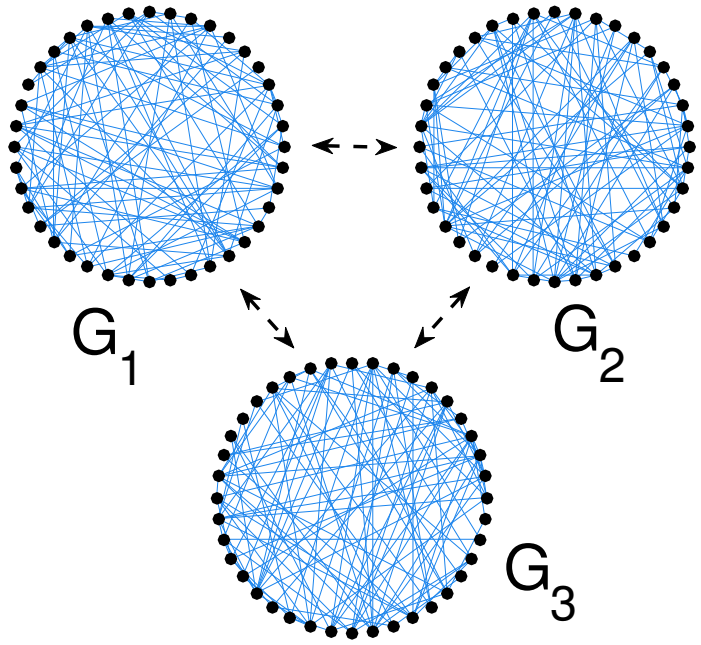

Figure 3.4: The underlying graphs of the network in Example 3.5.7

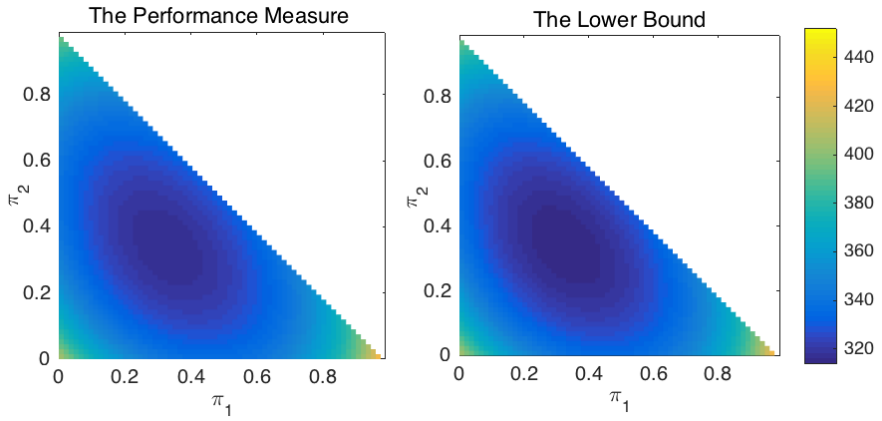

Figure 3.5: These colored plots show the performance measure and its lower-bound computed for $\pi_1, \pi_2 \in [0, 1]$, where $\pi_1 + \pi_2 + \pi_3 = 1$.

Running Time Analysis: The evaluation of the lower-bound (3.20) compared to the performance measure (3.13) turns out to have higher computational advantages for large networks. To compute (3.13), an eigenvalue decomposition of $\mathbf{E}_{\mathcal{W}} \in \mathbb{R}^{N^2 \times N^2}$ is required, which for a full matrix, in practice, is $O((N^2)^{2.8}) = O(N^{5.6})$, and for a sparse matrix $O(mN^2)$ [71], where m is the number of nonzero elements. However, for (3.20), there are algorithms that may compute the largest eigenvalue and its eigenvector in $O(m)$, If $\mathbf{E}_{\mathcal{W}}$ is not sparse, the

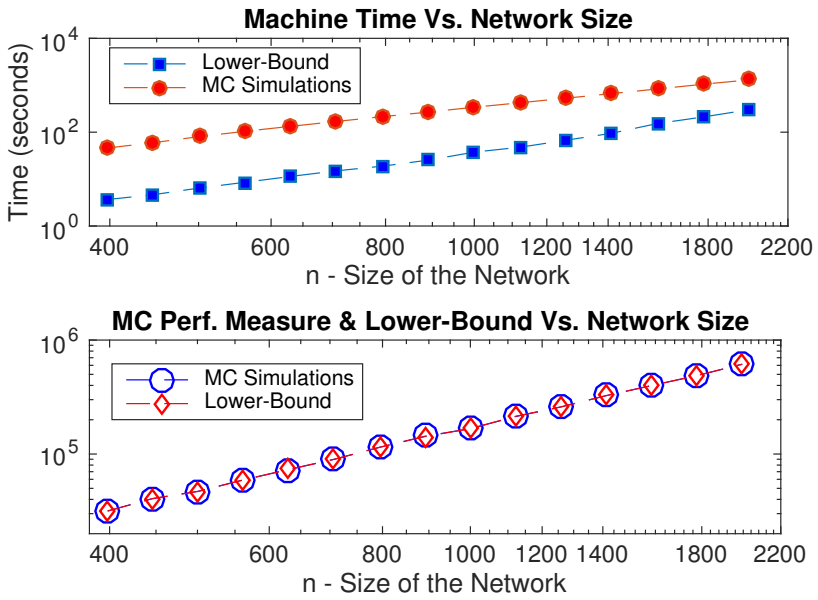

Figure 3.6: The lower-bound compared to the value of performance measure derived from MC method, together with the machine times.

time-complexity would be $O(N^4)$.

In an example, we examine the empirical relation between the network size and the running time.

Example 3.5.9 (Time vs. Size). Choosing $N \in [6, 60]$, we create 100 pairs of ER graphs with $p = 3\log(N)/N$ and assess the performance measure and its lower-bound for $\pi_1 \in [0, 1]$ (6 points). We record the average machine times and the results are summarized in Fig. 3.7. For large N, the time for the measure hikes considerably faster compared to the growth rate of the time required for evaluating the lower-bound.

3.6 Design for Optimal Random Switching

In the rest of this chapter, we introduce and characterize the problem of minimizing the performance measure by optimal design of the probability vector Π.

Problem 3.6.1 (Optimal Random Switching). *Given a set of networks $R \subset \mathcal{S}$ and cost vector $\mathcal{C} := [\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m]^T \succ 0$, find a probability mass function $\pi(\mathbf{W}_i) = \pi_i$ for all $\mathbf{W}_i \in R$, stacked as a vector $\Pi = (\pi_1, \pi_2, \ldots, \pi_m)^T$, minimizing a combination of the*

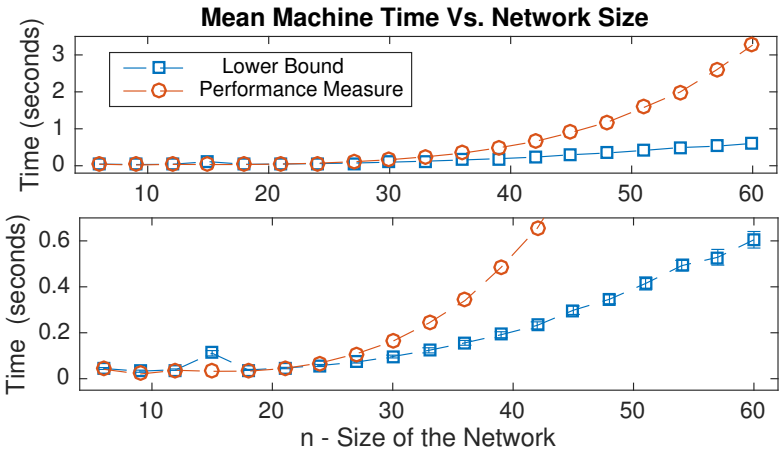

Figure 3.7: The mean time of the computations versus number of the nodes n (the lower figure magnifies the details of the upper figure)

performance measure and cost of the operation; i.e. for some augmenting coefficient $\gamma \geq 0$, *solve*

$$\underset{\Pi}{\text{minimize}} \quad \rho(\mathcal{W}) + \gamma \Pi^T \mathcal{C} \tag{3.25}$$

$$\text{subject to} : \Pi \succeq 0 \quad \text{and} \quad \Pi^T \mathbf{1}_m = 1.$$

Next, we reshape (3.25) as a tractable convex program; i.e. a Semi-Definite Program (SDP) [72], provided that the output matrix is $\mathbf{M}_N$.

Theorem 3.6.2. *Optimization Problem (3.6.1) with* $\mathbf{C} = \mathbf{M}_N$ *is equivalent to*

$$\underset{\Pi,\ t}{\text{minimize}} \quad t \tag{3.26}$$

$$\text{subject to} : \Pi \succeq 0,\ \Pi^T 1_m = 1 \text{ and}$$

$$\begin{pmatrix} \mathbf{I}_{N^2} - \mathbf{G}_{\mathcal{W}} & \text{vec}(\mathbf{M}_N) \\ \text{vec}(\mathbf{M}_N)^T & t - \gamma \Pi^T \mathcal{C} \end{pmatrix} \succeq \mathbf{0}.$$

This convex reformulation implies that we can directly look at the convexity of the performance measure in variable Π. Our next result magnifies this point.

Theorem 3.6.3. *If* $\rho(\mathbf{G}_{\mathcal{W}}) < 1$ *and* $\mathbf{C} = \mathbf{M}_N$, *the performance measure* $\rho(\mathcal{W})$ *is a convex*

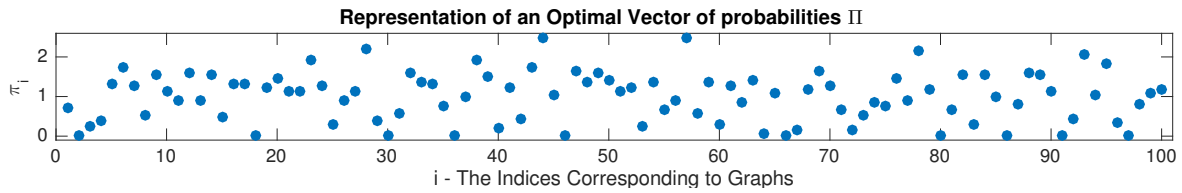

Figure 3.8: Representation of an optimal vector of probabilities Π for a network with $m = 100$ underlying graphs.

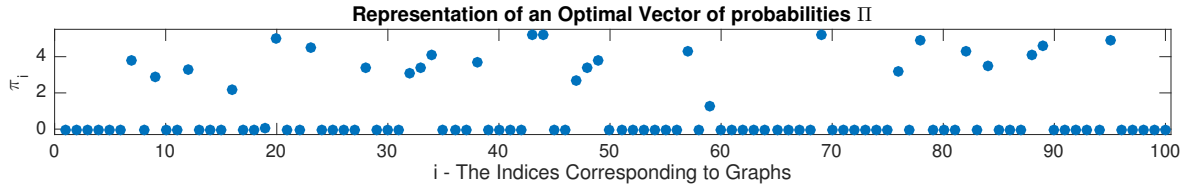

Figure 3.9: Representation of a sparse optimal Π with reweighted ℓ_1 penalization, where more entries are on the horizontal axis.

function of the probability vector $\Pi = [\pi_1, \pi_2, \ldots, \pi_m]^T$.

Although we can directly verify the convexity of this problem, the mere fact that the optimization problem is equivalent to a SDP implies that the objective function is a convex function of the variable Π.

3.6.1 Scenarios for the Design Problem

Assume a hypothetical scenario of coordination where at each sampling time only a subset of communication links exist between the agents. One may think of different advantages for this: (i) the power-consuming continuous communications are no longer needed, (ii) the amount of information to be processed (i.e. computational burden of the operations on an agent) may decrease (iii) the control input magnitudes may shrinks as we consider partial synchronization at each step. The next example considers this scenario.

Example 3.6.4 (Mixing Disconnected Networks). Consider 100 different graphs over $N = 24$ nodes. Each graph has the same number of randomly chosen links $|\mathcal{E}_i| = 4$. An LTI consensus network corresponding to each graph has unbounded measure because each individual graph is disconnected. Using Theorem 3.6.2, we implement (3.26) with $\gamma = 0$ in CVX [39] to find an optimal probability vector Π. In Fig. 3.8, we show an optimal Π (in percent) found in 113 seconds.

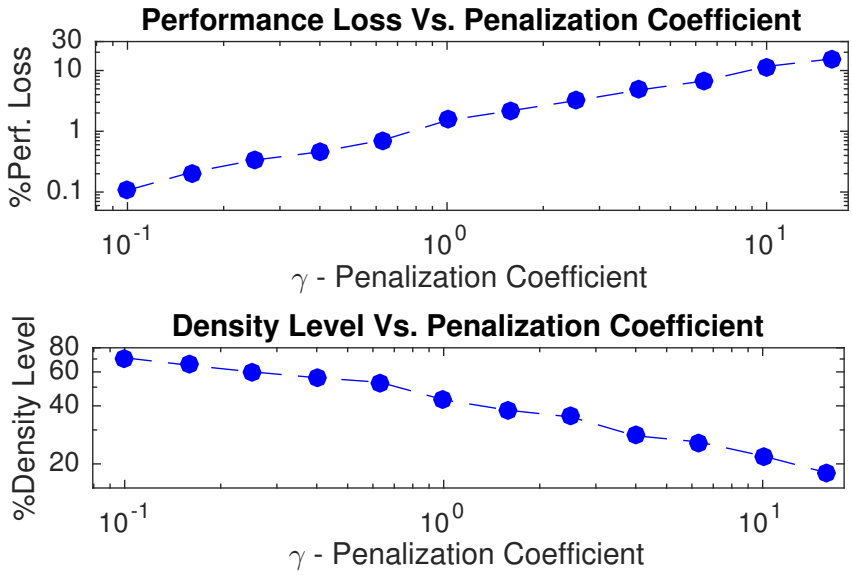

Figure 3.10: The Performance Loss and Density Level tradeoff illustrated by increasing the value of γ (see Example 3.6.6).

Example 3.6.5 (Sparse Selection of Disconnected Networks). The number of nonzeros in the optimal Π in Example 3.6.4 is high. As a remedy, we use Reweighted ℓ_1-Penalization [73]. To do so, we have a vector of weights at iteration j, $w^j \in \mathbb{R}^m$ is initially $w^1 = \mathbf{1}_m$. In our problem, we set $\mathcal{C} = w^j$. Then, at each iteration we compute the optimal solution $\Pi^j = [\pi_1^j, \pi_2^j, \ldots, \pi_m^j]$ of (3.26) with $\gamma = 5$. For the next iteration, we update the weights according to

$$w^{j+1}(i) = \frac{1}{e + \pi_i^j}, \text{ for all } i \in \{1, 2, \ldots, m\},$$

with $e = 10^{-6}$. We display the elements of the optimal value (in percent) in Fig. 3.9 after 5 iterations. Due to this penalization, the density level (i.e, the percent of nonzero elements of π) dropped from 90% to 26%, and the performance measure grew from around 852 to 902, a loss of around 6%.

An influential parameter in Example 3.6.5 is γ. On one side of the spectrum, $\gamma = 0$ implies the lack of requirement on the sparsity of Π. However, as γ increases, the sparsity is more favored, and we expect observing more performance loss. Our final example reflects this tradeoff.

Example 3.6.6 (Sparsity-Performance Tradeoff). Reconsider the settings of Example 3.6.5.

For $\gamma \in [0.1, 15]$, we examine the density level of the optimal vector Π and the performance measure at the same time and the performance loss and density level versus γ appear in Fig. 3.10.

3.6.2 Additional Convex Attributes of the Performance Measure

Theorem 3.6.3 asserts that $\rho(\mathcal{W})$ is a convex function of the probability vector Π. Here, we note that it is also a convex function of the stacked matrix of all $\mathbf{W}_i \in R$. Thus, first we introduce a stacked matrix

$$\mathbf{W}_{st} := \Big[\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_m\Big]^T, \text{ for all } \mathbf{W}_i \in R.$$

Proposition 3.6.7. *If $\lambda_{\max}(\mathbf{G}_{\mathcal{W}}) < 1$, the performance measure (4.18) is a convex function of stacked matrix $\mathbf{W}_{st}$.*

Next, we point out that the performance measure inherits this convexity as a function of the weights of all graphs of the switching network. For graph i with edge set $\mathcal{E}_i$, let $E_i := \text{vec}(w_{jk})$ (for all $\{j,k\} \in \mathcal{E}_i$) be a vector representing the weights of the corresponding graph. Stacking these vectors for all $\mathbf{W}_i \in \mathcal{W}_m$, we build a vector of weights

$$E_{st} := \left[E_1^T, E_2^T, \ldots, E_m^T\right]^T.$$

Theorem 3.6.8. *If for all $\mathbf{W}_i \in R$, $\mathbf{W}_i = \mathbf{I}_N - \epsilon_i \mathbf{L}_i \in \mathcal{S}$ for some graph Laplacian $\mathbf{L}_i$ and a proper $\epsilon_i > 0$, then the performance measure (4.18) is a convex function of E_{st}.*

The notion of convexity with respect to the weights of the graph is useful for design or redesign purposes, since numerous convex optimization techniques and their operational guarantees could be potentially considered.

3.7 Discussion and Future Directions

We characterized different aspects of the performance measure of randomly switching consensus networks that are perturbed by a Gaussian noise. In the first half of the Chapter, a

practical lower-bound on the measure was introduced, together with an analysis regarding the time-complexities and errors. A caveat might be the high order of time-complexity for the evaluation of the lower-bound, although it was cheaper than the performance measure. This facet may be investigated as a future work. Another concern is that the approximation bound for the lower-bound derived in Theorem 3.5.5 may not be tight, although the lower-bound is observed to be tight. When it comes to the optimization procedures (e.g. in topology design problems), if the computation of the measure compared to the lower-bound measure is expensive (that is indeed true for large networks), the bound may be a handy tool for the derivative-free optimization methods [74], which usually require cheap function evaluations. This is another potential direction for the further investigation.

In the second part of this manuscript, we defined the problem of optimal random switching and reformulated it as an SDP The objective is to find an optimal probability vector to achieve superior performance by mixing two or more consensus networks of inferior (or even unbounded) performance measures. However, the scalability of the current numerical scheme is rather questionable. We also proved the convexity of the performance measure in terms of the vector of probabilities and also in terms of simultaneously all weights of all graphs. This turns out to be important for topology design problems.

Appendix: Proofs

Proof of Theorem 3.4.1. Let us define the $z(t) := \mathbf{M}_N x(t)$ and $\mathbf{W}_c := \mathbf{M}_N \mathbf{W}$. If $\mathbf{W} \in \mathcal{S}$ and $\rho(\mathbf{M}_N \mathbf{W}) < 1$, then the steady-state covariance matrix

$$\mathbf{Z} := \lim_{t \to \infty} \mathbb{E}\left\{ z_t z_t^T \right\}$$

which has been shown to have the unique value of

$$\mathbf{Z} = \left(\mathbf{I}_N - \mathbf{W}_c^2 \right)^{-1} - \mathbf{J}_N / N,$$

(see [69]). The steady-state covariance matrix $\mathbf{Y}$ of $y_t = \mathbf{C} z_t$ is $\mathbf{Y} = \mathbf{C}\mathbf{Z}\mathbf{C}^T = CZC$. Since $\mathbf{C}\mathbf{J}_N = \mathbf{0}$, $\mathbf{Y} = \mathbf{C}(\mathbf{I}_n - \mathbf{W}_c^2)^{-1}\mathbf{C}$ is resulted. Because $\rho(\mathbf{W}) = \mathrm{Tr}(\mathbf{Y})$, $\mathbf{C}^2 = \mathbf{L}$ and

$\text{Tr}(\mathbf{ABC}) = \text{Tr}(\mathbf{CAB})$ one can verify (3.10). If $\mathbf{C} = \mathbf{L} = \mathbf{M}_N$,

$$\rho(\mathbf{W}) = \text{Tr}\left((\mathbf{I}_N - \mathbf{J}_N/N) \left(\mathbf{I}_N - \mathbf{M}_N \mathbf{W}^2\right)^{-1} \right),$$

which can be expanded to express the performance measure as

$$\rho(\mathbf{W}) = \text{Tr}\left((\mathbf{I}_N - \mathbf{M}_N \mathbf{W}^2)^{-1} - (\mathbf{J}_N/N)(\mathbf{I}_N - \mathbf{M}_N \mathbf{W}^2)^{-1} \right). \tag{3.27}$$

Since $\lambda_{\max}(\mathbf{M}_N \mathbf{W}) < 1 (\mathbf{M}_N \mathbf{W}) < 1$, using a Neumann series, we may write the second term as

$$\frac{\mathbf{J}_N}{N}(\mathbf{I}_N - \mathbf{M}_N \mathbf{W}^2)^{-1} = \frac{\mathbf{J}_N}{N} \sum_{i=0}^{\infty} (\mathbf{M}_N \mathbf{W}^2)^i = \mathbf{J}_N/N.$$

Because $\text{Tr}(\mathbf{J}_N/N) = 1$, we arrive at (3.11), which has been shown in [69] that is equivalent to (3.12). □

Proof of Theorem 3.4.2. We define $z(t) := \mathbf{M}_N x(t)$ and $\tilde{\mathbf{W}}(t) := \mathbf{M}_N \mathbf{W}(t)$. One inspects that the following chain of identities hold.

$$\mathbf{M}_N \mathbf{W}(t) = \mathbf{M}_N \mathbf{M}_N \mathbf{W}(t) = \mathbf{M}_N \mathbf{W}(t) \mathbf{M}_N.$$

We can now show that the dynamics of $z(t)$ are given by

$$z(t+1) = \tilde{\mathbf{W}}(t) z(t) + \mathbf{M}_N \xi(t),$$

which helps evaluate its outer product $\mathbf{Z}(t) := z(t) z(t)^T$ as

$$\mathbf{Z}(t+1) = \tilde{\mathbf{W}}(t) \mathbf{Z}(t) \tilde{\mathbf{W}}(t) + \mathbf{M}_N \xi(t) \xi(t)^T \mathbf{M}_N + 2\tilde{\mathbf{W}}(t) z(t) \xi(t)^T \mathbf{M}_N.$$

Taking the expected value, since $\xi(t)$ is independent of $z(t)$, the last term vanishes. Using Assumption 3.3.2, we may write $\mathbb{E}\{\xi(t)\xi(t)^T\} = \mathbf{I}_N$ to obtain

$$\mathbb{E}\{\mathbf{Z}(t+1)\} = \mathbb{E}\{\tilde{\mathbf{W}}(t) \mathbf{Z}(t) \tilde{\mathbf{W}}(t)\} + \mathbf{M}_N,$$

where we have used $\mathbf{M}_N\mathbf{M}_N = \mathbf{M}_N$. The vectorization of the last identity results in

$$\mathbb{E}\{\text{vec}(\mathbf{Z}(t+1))\} = \mathbb{E}\{\tilde{\mathbf{W}}(t) \otimes \tilde{\mathbf{W}}(t)\text{vec}(\mathbf{Z}(t))\} + \text{vec}(\mathbf{M}_N).$$

Using the conversion identity between the Kronecker product and vectorization, the independence of $\tilde{\mathbf{W}}(t)$ and $Z(t)$, and definition of $\mathbf{G}_{\mathcal{W}}$, we end up with a linear time-invariant dynamical system

$$\mathbb{E}\{\text{vec}(\mathbf{Z}(t+1))\} = \mathbf{G}_{\mathcal{W}}\mathbb{E}\{\text{vec}(\mathbf{Z}(t))\} + \text{vec}(\mathbf{M}_N), \tag{3.28}$$

that is stable if and only if $\lambda_{\max}(\mathbf{G}_{\mathcal{W}}) < 1$. If that is the case, then

$$\mathbf{Z}_{ss} := \lim_{t\to\infty} \mathbb{E}\{\text{vec}(\mathbf{Z}(t)^T)\}$$

is the unique solution to linear system of equations

$$\mathbf{Z}_{ss} = \mathbf{G}_{\mathcal{W}}\mathbf{Z}_{ss} + \text{vec}(\mathbf{M}_N).$$

Because $\lambda_{\max}(\mathbf{G}_{\mathcal{W}}) < 1$, its only solution is

$$\mathbf{Z}_{ss} = (\mathbf{I}_{N^2} - \mathbf{G}_{\mathcal{W}})^{-1}\text{vec}(\mathbf{M}_N). \tag{3.29}$$

Since $y(t) = \mathbf{C}z(t) = \mathbf{M}_N\mathbf{C}z(t)$ we compute

$$\mathbf{Y}_{ss} := \lim_{t\to\infty} \mathbb{E}\{\text{vec}(y(t)y(t)^T)\},$$

which is equivalently given by

$$\mathbf{Y}_{ss} = \lim_{t\to\infty} \mathbb{E}\{(\mathbf{M}_N\mathbf{C} \otimes \mathbf{M}_N\mathbf{C})\,\text{vec}(\mathbf{Z}(t))\},$$

which can be combined with (3.29) to obtain

$$\mathbf{Y}_{ss} = (\mathbf{M}_N \otimes \mathbf{M}_n)\, \mathbf{K}(\mathbf{I}_{N^2} - \mathbf{G}_{\mathcal{W}})^{-1}\mathrm{vec}(\mathbf{M}_N).$$

It holds that $\rho(\mathcal{W}) = \mathrm{vec}(\mathbf{I}_N)^T \mathbf{Y}_{ss}$. Therefore, we can write

$$\mathrm{vec}(\mathbf{I}_N)^T\, (\mathbf{M}_N \otimes \mathbf{M}_N) = \mathrm{vec}(\mathbf{M}_N \mathbf{I}_N \mathbf{M}_N)^T = \mathrm{vec}(\mathbf{M}_N)^T,$$

which verifies (3.13). Finally, because $\mathrm{vec}(\mathbf{M}_N)^T \mathbf{K} = \mathrm{vec}(\mathbf{C}\mathbf{M}_N\mathbf{C}^T)^T = \mathrm{vec}(\mathbf{L})^T$ we have verified the claim (3.14). □

Proof of Theorem 3.5.3. Symmetry of (3.17) gives $\hat{\mathbf{W}} := \mathbf{B}^* = \mathbf{C}^*$. Proposition 3.5.2 implies that $\mathbf{W}_d$ is a linear combination of $\mathbf{W}_i$'s, because based on (3.17), $\mathbf{E}_{\mathcal{W}}$ is a sum of Kronecker products. Since $\mathbf{1}_n$ is an eigenvector of every $\mathbf{W}_i \in R$, so it is an eigenvector of a linear combination of them $\hat{\mathbf{W}}$. Part (2) of Proposition 3.5.2 implies that

$$\hat{\mathbf{W}} \otimes \hat{\mathbf{W}} \preccurlyeq \mathbf{E}_{\mathcal{W}}.$$

We multiply both sides by $(\mathbf{M}_N \otimes \mathbf{M}_N) \succcurlyeq \mathbf{0}$ to get

$$\mathbf{M}_N\hat{\mathbf{W}} \otimes \mathbf{M}_N\hat{\mathbf{W}} \preccurlyeq \mathbf{G}_{\mathcal{W}}.$$

The spectral radius of the left hand side is $\lambda^2_{\max}(\mathbf{M}_N\hat{\mathbf{W}})$, thus

$$\lambda^2_{\max}(\mathbf{M}_N\hat{\mathbf{W}}) \leq \lambda_{\max}(\mathbf{G}_{\mathcal{W}}) < 1,$$

or $\rho(\mathbf{M}_N\hat{\mathbf{W}}) < 1$. It holds that $\mathbf{M}_N\hat{\mathbf{W}} = \mathbf{M}_N\mathbf{W}_s$ so $\rho(\mathbf{M}_N\mathbf{W}_s) < 1$. Based on Proposition 3.5.2, $\hat{\mathbf{W}}$ and, consequently, $\mathbf{W}_s$ are positive-definite. Also, $\lambda_1(\mathbf{W}_s) = 1$ with $v_1 = \mathbf{1}_n$. Hence, $\mathbf{W}_s \in \mathcal{S}$.

Based on the closed-form (3.13), we have $\rho(\mathcal{W}) = m_N^T \mathcal{Q} m_n$, where $m_n := \mathrm{vec}(\mathbf{M}_N)$. If we define

$$\mathcal{Q}_d := \mathbf{K}(\mathbf{I}_{N^2} - (\mathbf{M}_N\hat{\mathbf{W}}) \otimes (\mathbf{M}_N\hat{\mathbf{W}}))^{-1},$$

we can use $\mathbf{M}_N\hat{\mathbf{W}} = \mathbf{M}_N\mathbf{W}_s$ to show that

$$\mathcal{Q}_d := \mathbf{K}(\mathbf{I}_{N^2} - (\mathbf{M}_N\mathbf{W}_s) \otimes (\mathbf{M}_N\mathbf{W}_s))^{-1}.$$

Hence $\rho_{ss}(\mathbf{W}_s) = m_n^T\mathcal{Q}_d m_n$, where $\rho(\mathbf{M}_N\mathbf{W}_s) < 1$.

Since $\mathbf{G}_{\mathcal{W}} \succcurlyeq \mathbf{M}_N\hat{\mathbf{W}} \otimes \mathbf{M}_N\hat{\mathbf{W}}$, we can write

$$(\mathbf{I}_{N^2} - \mathbf{G}_{\mathcal{W}})^{-1} \succcurlyeq (\mathbf{I}_{N^2} - \mathbf{M}_N\hat{\mathbf{W}} \otimes \mathbf{M}_N\hat{\mathbf{W}})^{-1},$$

by $(\mathbf{I}_{N^2} - \mathbf{G}_{\mathcal{W}})^{-1}$, $(\mathbf{I}_{N^2} - \mathbf{M}_N\hat{\mathbf{W}} \otimes \mathbf{M}_N\hat{\mathbf{W}})^{-1} \succ 0$ (Because $\lambda_{\max}(\mathbf{G}_{\mathcal{W}}), \lambda_{\max}(\mathbf{M}_N\hat{\mathbf{W}}) < 1$). Multiplying by $\mathbf{K} \succcurlyeq \mathbf{0}$ gives

$$\mathbf{K}(\mathbf{I}_{N^2} - \mathbf{G}_{\mathcal{W}})^{-1} \succcurlyeq \mathbf{K}(\mathbf{I}_{N^2} - \mathbf{M}_N\hat{\mathbf{W}} \otimes \mathbf{M}_N\hat{\mathbf{W}})^{-1},$$

implying $\mathcal{Q} \succcurlyeq \mathcal{Q}_d$. Hence, $m_n^T\mathcal{Q}m_n \geq m_n^T\mathcal{Q}_d m_n$, which proves the claim. □

Proof of Theorem 3.5.5. We can evaluate $\Delta\rho = \mathrm{Tr}(\Delta\rho)$ as

$$\Delta\rho = \mathrm{vec}(\mathbf{L})^T\Delta^{-1}\mathbf{G}\mathrm{vec}(\mathbf{M}_N).$$

Let $\mathbf{R}_1 := \mathrm{vec}(\mathbf{M}_N)\mathrm{vec}(\mathbf{L})^T$, whose rank is one. We use the von Neumann's inequality: If $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times n}$ with ordered singular values $\sigma_1 \leq \ldots \leq \sigma_n$, then it holds that

$$|\mathrm{Tr}(\mathbf{AB})| \leq \sum_{i=1}^{n} \sigma_i(\mathbf{A})\sigma_i(\mathbf{B}).$$

If $\mathbf{A}$ is rank-one, then this boils down to $|\mathrm{Tr}(\mathbf{AB})| \leq \|A\|_2\|B\|_2$. Thus,

$$\Delta\rho = \mathrm{Tr}(\mathbf{R}_1\Delta^{-1}\mathbf{G}) \leq \|\mathbf{R}_1\|_2\|\Delta^{-1}\mathbf{G}\|_2. \tag{3.30}$$

For a matrix $\mathbf{A}$ perturbed by a small matrix $\Delta\mathbf{A}$, the matrix inversion has the following first order error bound.

$$\|(\mathbf{A} + \Delta\mathbf{A})^{-1} - \mathbf{A}^{-1}\|_2 \leq \|\mathbf{A}^{-1}\|_2^2\|\Delta\mathbf{A}\|_2.$$

(e.g. see [75]). It can be shown that this is equivalent to

$$\|(\mathbf{A}+\Delta\mathbf{A})^{-1}-\mathbf{A}^{-1}\|_2 \leq \|(\mathbf{A}+\Delta\mathbf{A})^{-1}\|_2^2\|\Delta\mathbf{A}\|_2.$$

For the last term of (3.30) this inversion error implies

$$\|\Delta^{-1}\mathbf{G}\|_2 \leq \|(\mathbf{I}_{N^2}-\mathbf{M}_N\hat{\mathbf{W}}\otimes\mathbf{M}_N\hat{\mathbf{W}})^{-1}\|_2^2\|\Delta\mathbf{A}\|_2, \tag{3.31}$$

where the right hand side, using the fact that $\|.\|_2=\lambda_{\max}(.)$ for symmetric real matrices, is equal to

$$\frac{\|\Delta\mathbf{G}\|_2}{(1-\lambda_{\max}^2(\mathbf{M}_N\hat{\mathbf{W}}))^2} \leq \frac{\|\Delta\mathbf{G}\|_F}{(1-\lambda_{\max}^2(\mathbf{M}_N\hat{\mathbf{W}}))^2}, \tag{3.32}$$

where we have used that $\|.\|_2\leq\|.\|_F$. Combining $\|\Delta\mathbf{G}\|_F\leq c_m$, (3.30), (3.31), and (3.32) altogether verifies (3.23). □

Proof of Proposition 3.6.7. First, we need the following lemma.

Lemma 3.7.1. *Define* $g:\mathbb{R}^{Nm\times m}\to\mathbb{R}$ *as*

$$g(\mathbf{X})=c^T\mathbf{A}\left(\mathbf{I}_{N^2}-\mathbf{B}h(\mathbf{X})\right)^{-1}c, \tag{3.33}$$

where $\mathbf{X}:=\Big[\mathbf{X}_1,\mathbf{X}_2,\ldots,\mathbf{X}_m\Big]^T$, $c\in\mathbb{R}^{N^2}$, $\mathbf{A},\mathbf{B}\in\mathbb{R}^{N^2\times N^2}$, *symmetric and* $\mathbf{A},\mathbf{B}\succcurlyeq\mathbf{0}$, *and*

$$h(\mathbf{X}):=\sum_{i=1}^{m}\mathbf{X}_i\otimes\mathbf{X}_i.$$

Then, for all $N\times N$ *symmetric matrices* $\mathbf{X}_i\succ 0$ *satisfying* $\lambda_{\max}\left(\mathbf{B}h(\mathbf{X})\right)<1$, $g(\mathbf{X})$ *is a convex function of* $\mathbf{X}$.

Proof of Lemma 3.7.1. Let $\mathbf{F},\mathbf{V}\in\mathbb{R}^{Nm\times m}$ be

$$\mathbf{F}=\Big[\mathbf{F}_1,\mathbf{F}_2,\ldots,\mathbf{F}_N\Big]^T$$

and

$$\mathbf{V} = \left[\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_N\right]^T,$$

where $\mathbf{F}_i, \mathbf{V}_i \succ 0$ and $\lambda_{\max}\left(\mathbf{B}h(\mathbf{F}+t\mathbf{V})\right) < 1$. It holds that $g(\mathbf{X})$ is convex if and only if

$$g(t) := g(\mathbf{F}+t\mathbf{V})$$

is convex [72]. We can write $g(t) = c^T\mathbf{A}\mathbf{D}(t)^{-1}c$, where $\mathbf{D}(t) := \mathbf{I}_{N^2} - \mathbf{B}h(\mathbf{F}+t\mathbf{V})$. By the chain rule we get

$$\frac{d^2g(t)}{dt^2} = c^T\mathbf{A}\mathbf{D}^{-1}(t)\left(\mathbf{H}(t) - \frac{d^2\mathbf{D}}{dt^2}\right)\mathbf{D}^{-1}(t)c,$$

where $\mathbf{H}(t)$ is given by

$$\mathbf{H}(t) := 2\frac{d\mathbf{D}}{dt}\mathbf{D}^{-1}(t)\frac{d\mathbf{D}}{dt}.$$

Since $\mathbf{D}^{-1}(t) \succcurlyeq 0$ and $d\mathbf{D}/dt$ is symmetric, $\mathbf{H}(t) \succcurlyeq \mathbf{0}$. Note that $\mathbf{D}(t)$ is given by

$$\mathbf{D}(t) = \mathbf{I}_{N^2} - \mathbf{B}\left(h(\mathbf{F}) + t^2h(\mathbf{V}) + \sum_{i=1}^{N} t\mathbf{F}_i \otimes \mathbf{V}_i + t\mathbf{V}_i \otimes \mathbf{F}_i\right).$$

Therefore, we have

$$\frac{d^2\mathbf{D}}{dt^2} = -\sum_{i=1}^{N} 2\mathbf{B}(\mathbf{V}_i \otimes \mathbf{V}_i) \preccurlyeq \mathbf{0},$$

and $\mathbf{H}(t) - d^2\mathbf{D}/dt^2 \succcurlyeq \mathbf{0}$. Because $\mathbf{D}^{-1}(t)$ is symmetric, we get

$$\mathbf{D}^{-1}(t)\left(\mathbf{H}(t) - \frac{d^2\mathbf{D}}{dt^2}\right)\mathbf{D}^{-1}(t) \succcurlyeq \mathbf{0},$$

which can be multiplied by $\mathbf{A} \succcurlyeq \mathbf{0}$, hence

$$\mathbf{Q}(t) := \mathbf{A}\mathbf{D}^{-1}(t)\left(\mathbf{H}(t) - \frac{d^2\mathbf{D}}{dt^2}\right)\mathbf{D}^{-1}(t) \succcurlyeq \mathbf{0}.$$

So $d^2g(t)/dt^2 = c^T\mathbf{Q}(t)c \geq 0$, and $g(\mathbf{X})$ is convex. □

Now, in Lemma 3.7.1, let $\mathbf{X}_i = \sqrt{\pi_1}\mathbf{W}_i$ for $\mathbf{W}_i \in R$, $c = \mathrm{vec}(\mathbf{M}_N)$, $\mathbf{A} = \mathbf{C} \otimes \mathbf{C}$, $\mathbf{B} = \mathbf{M}_N \otimes \mathbf{M}_N$. $\mathbf{X}_i \succ \mathbf{0}$, $\mathbf{A}, \mathbf{B} \succcurlyeq \mathbf{0}$. Also, $\lambda_{\max}\left(\mathbf{B}h(\mathbf{X})\right) = \lambda_{\max}(\mathbf{G}_{\mathcal{W}}) < 1$. Thus, using Lemma 3.7.1 for ρ, the claim follows. □

Proof of Theorem 3.6.8. By definition, $\mathbf{L}_i$ is an affine map of E_i , thus $\mathbf{W}_i = \mathbf{I}_N - \epsilon\mathbf{L}_i$ is an affine map of E_i. Hence, $\mathbf{W}_{st}$ is an affine map of E_s. By Proposition 3.6.7, the performance measure is a convex function of E_s, since the composition of a convex function with an affine map preserves the convexity [72]. □

Chapter 4

Performance of a Class of Nonlinear Consensus Networks

4.1 Introduction

The central objective in the theory of networked control systems is to address and analyze the practical challenges in implementations of real-world dynamical networks, in order to develop design algorithms with certified convergence properties [12, 14, 76–81]. The application areas, nowadays, range from multi-robot systems [82] to social networks [83], power systems [84], metabolic pathways [85–87], and brain networks [88]. One of the inherent unappealing features of these real-world networks is the nonlinearity of the interactions among the subsystems that stem from how subsystems affect each other's dynamics [14, 89–93]. For example in the natural networks, physical interactions such as fluid field coupling [94], coupled biochemical reactions [87], or visual coordination [91] may result in nonlinear coupling among the subsystems.

The main focus of the existing body of literature is on stability analysis of nonlinear dynamical networks, where some of these works investigate effects of coupling topologies [84, 89], time-delay [95–97] and exogenous noise [98]. The common approach to deal with the existing nonlinearities is to study linearized forms of network dynamics. There is a rich number of works devoted to performance and robustness analysis and optimal design

of linear dynamical networks [1, 21, 26, 34, 46, 47, 52, 53, 55, 99–104]. Despite a growing need to analyze and synthesize the nonlinear dynamical networks in non-equilibrium modes of operation, consistent and systematic methods to tackle these problems are sorely missing in the literature. The main reason is that the linear network techniques, which are mainly based on eigendecomposition, cannot be applied to nonlinear systems. Recent advances in analysis of dynamical systems using Koopman operator theory have opened up a new venue to study the properties of nonlinear systems in a systematic manner [**?**, 105–108].

In this chapter, we build upon concepts and tools from Koopman methodology to assess the performance of a class of nonlinear consensus networks. These networks are defined over an undirected state-dependent interconnection graph topology, where the control input of each agent is equal to a weighted combination of the difference between its own state and its neighbors. The expected value of the output energy of the network is adopted as the performance measure. We obtain a closed-form series representation for this quadratic performance measure and show that the value of performance measure depends on the spectra of the Koopman operator. The idea of spectral characterization of performance measure can be potentially utilized to analyze and design nonlinear networks; we refer to [1,55,102,104] for successfulness of this approach in the case of linear dynamical networks. An efficient numerical algorithm is developed to compute the value of the performance measure for a given dynamical network. Several analytical and numerical examples have been provided to highlight the usefulness of our theoretical findings.

4.2 Preliminaries

Consider an autonomous dynamical system given by

$$\dot{x} = F(x), \tag{4.1}$$

with $F(x) : \mathbb{R}^n \to \mathbb{R}^n$ representing a C^2 vector field on $\mathbb{R}^n$. For the initial condition $x_0 \in \mathbb{R}^n$, $x(t) := S(t, x_0) : \mathbb{R}_+ \times \mathbb{R}^n \to \mathbb{R}^n$ is the generated flow of (4.1), which is assumed to be defined for all $t \geq 0$. We assume that F attains a hyperbolic stable fixed point at the

origin. i.e., $F(0) = 0$. Moreover, we denote the Jacobian of F at the fixed point by

$$\mathbf{A} := \frac{\partial}{\partial x} F|_{x=0}, \tag{4.2}$$

which we assume to be Hurwitz; i.e., the eigenvalues of $\mathbf{A}$ have strictly negative real parts. The basin of attraction of the origin is an open neighborhood of 0 with $\Omega \subset \mathbb{R}^n$ a compact subset of this neighborhood. By definition, $S(t, x_0) \in \Omega$ for any $x_0 \in \Omega$ and $t \geq 0$, such that $S(t, x_0) \to 0$ as $t \to +\infty$. Let us define the functional space

$$\mathcal{F} = \left\{ f \in C^1(\Omega, \mathbb{R}) : \sup_{x \in \Omega} \left| f(x) \right| + \sup_{x \in \Omega} \left\| \nabla f(x) \right\| < \infty \right\} \tag{4.3}$$

that together with norm $|f|_{C^1} := \sup_{x \in \Omega} \left| f(x) \right| + \sup_{x \in \Omega} \left\| \nabla f(x) \right\|$, constitute a Banach space. This will be the space of observable functions on flow $S(\cdot, x_0)$. For fixed $t \geq 0$, the Koopman operator $U^t : \mathcal{F} \to \mathcal{F}$ associated with (4.1) is

$$(U^t f)(x_0) = f \circ S(t, x_0). \tag{4.4}$$

For any fixed $t \geq 0$, it can be shown that U^t is linear in $\mathcal{F}$. Furthermore, the collection $\{U^t\}_{t \geq 0}$ constitutes a semigroup known as *the Koopman semigroup* [107]. In the context of continuous autonomous dynamical systems, (4.4) is interpreted as the action of semigroup on observable $f \in \mathcal{F}$. The spectrum of operator U^t may consist of a discrete, continuous and residual part. The discrete part, also known as point spectrum of U^t, is defined as

$$\sigma_p(U^t) = \left\{ \lambda \in \mathbb{C} \,\middle|\, U^t \phi = e^{\lambda t} \phi, \text{ for some } \phi = \phi_\lambda \in \mathcal{F} \right\}. \tag{4.5}$$

Throughout this chapter $\left(\lambda, \phi_\lambda\right)$, for $\lambda \in \sigma_p(U^t)$, is called the Koopman pair of an eigenvalue with its corresponding eigenfunction. The purpose of this work is to discuss the role of the Koopman operator theory in evaluating quadratic performance measures for a class of nonlinear consensus protocols that enjoy a great interest in the field of networked control systems. More specifically, we leverage a recent extension of the Hartman's theorem for hyperbolic dynamical systems [106] to outline the pivotal role of point spectrum in

approximating the output energy of nonlinear distributed cooperative algorithms.

The rest of the chapter is organized as follows. In Section 4.3, we will apply the extension of the Hartman's theorem in order to investigate the conditions under which one is able to express the flow $S(\cdot, x_0)$ of (4.1) in terms of the Koopman pairs, i.e., to write the i-th element of $S(\cdot, x_0)$ as

$$\big[S(t, x_0)\big]_i \approx \sum_{\lambda} c_{\lambda}^{(i)} \mathrm{e}^{\lambda t} \phi_{\lambda}(x_0), \qquad \text{for every } t \geq 0$$

for some coefficients $c_\lambda = [c_\lambda^{(1)}, \ldots, c_\lambda^{(n)}]^T$. Then, each collection $\{(\lambda, \phi_\lambda, c_\lambda)\}_\lambda$ will constitute a Koopman Mode Decomposition (KMD) [107]. We use an interesting fact about the map created by stacking specific eigenfunctions of the Koopman operator and its inverse map for dynamical systems with hyperbolic stable fixed points : polynomial approximations of the inverse map yields a Koopman Mode Decomposition.

Based on the results of Section 4.3, we proceed in Section 4.4 with the calculation of the performance measures for nonlinear consensus networks. The measures are expressed series form as a function of KMD's. In addition, we discuss a number of special cases where KMD's can be explicitly calculated.

In Section, 4.5 we describe a method to come-up with a sparse approximation to the eigenfunctions of the Koopman operator. The method strongly depends on a nearly-optimal fitting technique called Smolyak-Collocation projection. We use the same method to compute the approximate Koopman modes. Using the above developments, we may derive quantitative information about the stability and performance of nonlinear dynamical networks. In fact, , we look at the performance measure of a class of nonlinear dynamical systems and illustrate how their performance can be assessed using the spectra of the Koopman operator.

4.3 Koopman Mode Decomposition of System Flows

The celebrated theorem of Hartman (stated below for convenience) establishes a crucial connection between autonomous dynamical system (4.1) and the dynamics of the linearized

system around the origin. A moment of reflection, initially mentioned in [106], can lay the groundwork of bridging the gap between spectral properties of the nonlinear and the linearized system around the fixed point. The aim of the present section is to conduct a rigorous discussion of these exact steps. We begin our analysis with parts adapted from literature to keep the manuscript self-contained.

Theorem 4.3.1. *[Hartman's Theorem [109]] Consider dynamical system* (4.1) *with the smoothness assumptions on* $\mathbf{F}$ *to hold and the origin to be a hyperbolic fixed point. Then there exists a* C^1*-diffeomorphism* H *of a neighborhood* U *of the origin on an open set* $\Omega' \subset \Omega$ *containing the origin such that for each* $\mathbf{x}_0 \in \Omega'$*, there exists is an open interval* $I(\mathbf{x}_0) \subset \mathbb{R}_+$ *containing zero such that for all* $x_0 \in U$ *and* $t \in I(x_0)$

$$H \circ S(t, x_0) = \mathrm{e}^{\mathbf{A}t} H(x_0),$$

where $\mathbf{A} = \frac{\partial}{\partial \mathbf{x}} \mathbf{F}|_{\mathbf{x}=0}$.

Remark 12. The set $I(\mathbf{x}_0)$ stands for the maximal interval of existence of the solution of system (4.1), that defines flow $S(t, \mathbf{x}_0)$, for any $t \geq 0$. Evidently, $I(\mathbf{x}_0) = \mathbb{R}_+$ for all $\mathbf{x}_0$ in the basin of attraction Ω.

The next result extends the theorem of Hartman to hold true over the whole the basin of attraction of the fixed point at the origin.

Theorem 4.3.2. [106] *If* F *is* C^2 *and* $\mathbf{A} = \frac{\partial}{\partial \mathbf{x}} F|_{\mathbf{x}=0}$ *is Hurwitz, then there exists a diffeomorphism* $\alpha : \Omega \to \mathbb{R}^n$ *such that*

$$\alpha \circ \mathbf{S}(t, x_0) = \mathrm{e}^{\mathbf{A}t} \alpha(x_0), \tag{4.6}$$

for all $x_0 \in \Omega$ *and* $t \geq 0$.

Next, assuming that $\mathbf{A}$ is diagonalizable, we can write $\mathbf{A} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1}$ where $\mathbf{\Lambda}$ is a diagonal matrix, having diagonal elements with strictly negative real parts. Let us define

$$H(x) := \mathbf{R}^{-1} \alpha(x). \tag{4.7}$$

Then, one may observe that

$$H\left(S(t,x_0)\right) = \mathbf{R}^{-1}\mathrm{e}^{\mathbf{A}t}\mathbf{R}H(x_0) = \mathrm{e}^{\Lambda t}H(x_0). \tag{4.8}$$

Clearly, map $H : \Omega \to \mathbb{C}^n$ is a diffeomorphism. Hence, flow of the dynamical system $S(\cdot, x_0)$ can be expressed as

$$S(t,x_0) = H^{-1}\left(\mathrm{e}^{\Lambda t}H(x_0)\right), \quad \text{for every } t \geq 0 \text{ and } x_0 \in \Omega. \tag{4.9}$$

This suggests that knowledge of maps H and H^{-1} helps identify the flow of the system. In an interesting turn of events, there is an important correlation between Koopman spectrum and the eigenvalues of the Jacobian matrix $\mathbf{A}$ at the fixed point.

Theorem 4.3.3. *Let map H given in* (4.7) *have component-wise expression*

$$H = \left[H_1, H_2, \ldots, H_n\right]^T, \tag{4.10}$$

for $H_i : \Omega \to \mathbb{C}^n$ and $i = 1, \ldots, n$. If λ_i is the i-th eigenvalue of $\mathbf{A}$, then $\left(\lambda_i,\, H_i\right)$ is a pair of Koopman eigenvalue and its corresponding eigenfunction.

Proof. The result immediately follows after comparing the definition of the Koopman eigenfunction in (4.5) with identity (4.8). □

We take advantage of this connection to provide a Koopman Mode Decomposition (KMD) for dynamical systems with a stable hyperbolic fixed point. One may find the general aspects of this decomposition in [107]. In this context, Extended Dynamic Mode Decomposition (EDMD) [110] is a framework with focus on derivation of numerical estimations to Koopman operator and KMD. At first, we make two crucial remarks before coming up with the advertised decomposition.

Polynomial Expansion of H^{-1}. Clearly, all elements of $H^{-1}(x) = \alpha^{-1}(\mathbf{R}\,x)$ are continuous in Ω, hence they map compact sets onto compact sets. Therefore, the domain of definition of H^{-1} is compact.

By virtue of the Stone-Weierstrass Theorem [111] $H^{-1}(x)$ can be uniformly ϵ-approximated

over the domain of H^{-1} by multivariate polynomials. Therefore, for every $x \in \operatorname{dom} H^{-1}$ we can write

$$H^{-1}(x) \overset{\epsilon}{\approx} \sum_{\gamma \in \Gamma_\epsilon} c_\gamma^\epsilon \; x_1^{j_1} \cdots x_n^{j_n}, \tag{4.11}$$

where $\gamma = [j_1, \dots j_n]^T \in \mathbb{Z}_+^n$, $\overset{\epsilon}{\approx}$ implies the approximation with maximal error of ϵ, and $c_\gamma^\epsilon = c_{j_1,\dots j_n}^\epsilon$ is represented using the multi-index notation. The index set $\Gamma_\epsilon \subset \mathbb{Z}_+^n$ consists of finite number of indices based on the desired level of accuracy $\epsilon > 0$. If map H^{-1} is analytic, then it admits a Maclaurin expansion with a positive radius of convergence and we can have an infinite series representation (at least in a subset of Ω^{-1}) similar to (4.11).

Remark 13. Not every polynomial approximation of H^{-1} is suitable in this chapter. It is necessary for the right hand-side of (4.11) to vanish at the origin, as H^{-1} does as well. This property permits a credible polynomial approximation of the output energy of (4.1) in terms of Koopman modes. Examples of polynomial expansions that can approximate H^{-1} under such constraints are interpolation based methods using multi-variate polynomials of the Bernstein or Chebyshev families, with appropriate scaling of domain of H^{-1} [112].

Superposition of Koopman Eigenpairs. The closedness of the set of eigenfunctions under multiplication is an important property that is stated in the next lemma.

Lemma 4.3.4 ([107]). *Let $\phi_1, \phi_2 \in \mathcal{F}$ with associated eigenvalues λ_1 and λ_1, respectively. Then $\phi_3(x) := \phi_1(x)\phi_2(x) \in \mathcal{F}$ with associated eigenvalue $\lambda_3 = \lambda_1 + \lambda_2$*

We are ready now to formulate a KMD-based expression for flow $S(\cdot, x_0)$. For its exposition we consider an arbitrary but fixed ordering of the elements of $\mathbb{Z}_+^n$, $\mathbb{Z}_+^n = \{\gamma_1, \gamma_2, \dots, \gamma_i, \dots\}$ where $\gamma_i = (j_1, j_2, \dots, j_n)^T$.

Proposition 4.3.5. *Let $\mathbf{A} = \frac{\partial}{\partial x} F(x)|_{x=0}$ be diagonalizable and Hurwitz with eigenvalues $\lambda_1, \dots, \lambda_n$. Consider map $H^{-1}(x)$ with elements given (4.7) for every $x_0 \in \Omega$, where Ω is a compact set. Then, using the approximation (4.11) for $H^{-1}(x)$, flow $S(\cdot, x_0)$ of nonlinear*

system (4.1) *attains the representation*

$$S(t,x_0) \overset{\epsilon}{\approx} \sum_{i\geq 1} c_i \, \mathrm{e}^{\bar{\lambda}_i t}\phi_i(x_0), \qquad \textit{for all } t \geq 0$$

where for the ordered vector $\gamma_i = [j_1,\ldots,j_n]^T \in \Gamma_\epsilon \subset \mathbb{Z}_+^n$, *we have*

$$\bar{\lambda}_i := \sum_{k=1}^{n} j_k\lambda_k \quad \textit{and} \quad \phi_i(x_0) := \prod_{k=1}^{n} H_k^{j_k}(x_0). \tag{4.12}$$

Proof. Recall the expression (4.9) that is true for every $x_0 \in \Omega$. Substituting $\mathrm{e}^{\mathbf{\Lambda} t}H(\mathbf{x}_0)$ into (finite) series representation (4.11), we may write the flow of system (4.1) as

$$S(t,x_0) \overset{\epsilon}{\approx} \sum_{\gamma\in\Gamma_\epsilon} c_\gamma^\epsilon \ \left(\mathrm{e}^{\lambda_1 t}H_1(\mathbf{x}_0)\right)^{j_1} \ldots (\mathrm{e}^{\lambda_n t}H_n(\mathbf{x}_0))^{j_n}, \tag{4.13}$$

which can be reorganized to obtain

$$S(t,x_0) \overset{\epsilon}{\approx} \sum_{\gamma\in\Gamma_\epsilon} c_\gamma^\epsilon \ \mathrm{e}^{(j_1\lambda_1+j_2\lambda_2+\cdots+j_n\lambda_n)t}H_1^{j_1}(\mathbf{x}_0)\ldots H_n^{j_n}(\mathbf{x}_0).$$

Let us define $\bar{\lambda}_i$ and $\phi_i(x)$ according to (4.12). Using Lemma 4.3.4, we deduce that $\phi_i(x)$ is an eigenfunction of Koopman operator with eigenvalue $\bar{\lambda}_i$. Rewriting the flow and using the introduced notation gives us the desired representation. □

In fact, we derive the explicit decomposition introduced in Proposition 4.3.5 by extending the material presented in [105] or [106]. We will see that this decomposition is a necessary tool for the subsequent analysis. Before that, we recall a useful lemma, that identifies a partial differential equation to associate the Koopman pairs.

Lemma 4.3.6. *[See [105]] Consider a pair of Koopman eigenvalue and its corresponding eigenfunction denoted by* $\big(\lambda,\phi_\lambda(x)\big)$ *associated with nonlinear dynamics* (4.1). *The pair satisfies the identity*

$$F(x)^T\nabla\phi_\lambda(x) = \lambda\phi_\lambda(x). \tag{4.14}$$

4.4 Performance of Nonlinear Consensus Networks

The standard multi-agent setting regards a finite collection of agents labeled as $i = 1, 2, \ldots, n$. The i'th agent is characterized by a real-valued state x_i. In a consensus network with first order dynamics, the agents update their states by communicating with their adjacent (neighboring) agents. Our focus in this work is on the class of dynamic protocols of the form

$$\dot{x}_i = \sum_{\{i,j\}\in\mathcal{E}} w_{ij}\,(x_j - x_i), \tag{4.15}$$

where $\mathcal{E}$ is the set of edges of the undirected graph of the network whose weights are symmetric and state-dependent in the form of

$$w_{ij} = w_{ji} = \tilde{w}_{ij}\, g\left(|x_i - x_j|^2\right), \tag{4.16}$$

for $g : \mathbb{R}_+ \to \mathbb{R}_{++}$ a positive coupling function of the graph, and constant $\tilde{w}_{ij} > 0$. We note that such a state-dependence of the couplings is motivated by a natural assumption: the remote or dissimilar agents less likely interact with each other. For instance, this is the case in the context of social networks, oscillatory networks [89] or biological networks. For this reason function g is usually considered to be monotonically decreasing [79, 91]. By defining the state of the network as $x := [x_1, \ldots, x_n]^T \in \mathbb{R}^n$, we may express the collective dynamics of the agents as

$$\dot{x} = -\mathcal{L}_x\, x \tag{4.17}$$

where $\mathcal{L}_x$ is the state-dependent graph Laplacian matrix with coupling weights that vary according to (4.16). For subsequent analysis we rely on two conditions, stated right below.

Assumption 4.4.1. *The function g is analytic and it satisfies $g(0) = 1$.*

Assumption 4.4.2. *The graph with coupling weights $\{\tilde{w}_{ij}\}_{\{i,j\}\in\mathcal{E}}$ is connected.*

Connectedness implies that there exists a linked path between any two distinct nodes i and j in the graph of the network. A consequence of the latter assumption is that the graph

corresponding to $\mathcal{L}_x$ remains connected and undirected for all $x \in \mathbb{R}^n$, since $w_{ij} > 0$ for every $\{i, j\} \in \mathcal{E}$. The next result provides a standard sufficient condition for convergence of dynamical network (4.15) to consensus equilibrium.

Theorem 4.4.3. *Let Assumptions 4.4.1 and 4.4.2 hold true. For any initial state $x_0 \in \mathbb{R}^n$ the long term dynamics satisfy*

$$\lim_{t\to\infty} S(t, x_0) = \bar{x}\, \mathbf{1}_n,$$

where the average vector of the network is $\bar{x} := \frac{1}{n}\sum_{i=1}^{n} x_i(0)$. The convergence to consensus occurs exponentially fast, with a rate that depends on initial state x_0.

Proof. At first, observe that

$$\max_{i,j=1,\dots,n} |x_i(t) - x_j(t)| \leq \max_{i,j=1,\dots,n} |x_i(0) - x_j(0)| \text{ for all } t \geq 0.$$

This is easily verified since for the node i with the maximum initial condition $\max_i \dot{x}_i(t) \leq 0$. Similarly for the node i with the minimum starting value $\min_i \dot{x}_i(t) \geq 0$. The solution $\mathbf{x}(t, x_0)$ remains bounded in $\Omega_0 := [\min_i x_i(0), \max_i x_i(0)]$. Consider the Lyapunov functional $\Lambda(x) = \dfrac{1}{2}\sum_{i\neq j} |x_i - x_j|^2$. Then for the solution $\mathbf{x}(t),\ t \geq 0$ of (4.15), we have

$$\frac{d}{dt}\Lambda(x(t)) = \sum_{i\neq j} \big(x_i(t) - x_j(t)\big)\big(\dot{x}_i(t) - \dot{x}_j(t)\big) \leq -\beta(t)\Lambda(x(t))$$

where the value of $\beta(t)$ is given by

$$\beta(t) := \min_{\substack{s\in[0,t]\\ \{i,j\}\in\mathcal{E}}} w_{ij}\big(x(s)\big) \geq \underline{w} \cdot \underline{g} > 0$$

for $\underline{w} = \min_{i,j=1,\dots,n} \tilde{w}_{ij}$ and $\underline{g} = \min_{s_1,s_2\in\Omega_0} g(|s_1 - s_2|) > 0$ [113]. By virtue of graph connectivity the convergence to the agreement space $x_1 = x_2 = \cdots = x_n$, occurs exponentially fast. Finally, observe that $\frac{1}{n}\sum_{i=1}^{n} x_i$ is a first integral of motion to conclude about the consensus point. □

The average of x_0 is called the consensus equilibrium of the network over the state

of interest [14]. The central objective of this work is to evaluate systemic measures of performance that quantify the necessary effort the dynamical system takes to converge to consensus. We aim at leveraging the Koopman framework, developed in the previous section. The requirement for the implementation of that machinery is to have a hyperbolic and asymptotically stable fixed point. One may notice that

$$\mathbf{A} := -\mathcal{L}_0,$$

with a smallest eigenvalue in magnitude is $\lambda_1(\mathbf{A}) = 0$. Hence, the fixed point at the origin is not hyperbolic. In order to overcome this difficulty we introduce output dynamics vector y with elements $y_i := x_i - \frac{1}{n}\sum_{k=1}^{n} x_k$, or in matrix form, $y = \mathbf{M}_n x$, where $\mathbf{M}_n$ is the the centering matrix given by

$$\mathbf{M}_n := \mathbf{I}_n - \mathbf{J}_n/n \in \mathbb{R}^{n\times n},$$

where $\mathbf{J}_n$ is the square matrix of all ones. The dynamics of $\mathbf{y}$ constitute the disagreement network associated with (4.15) is defined to pass this obstacle [14, 26]. The disagreement Laplacian matrix is

$$\mathcal{L}_d(x) := \mathcal{L}_x + \frac{\delta}{n}\mathbf{J}_n$$

for some $\delta > 0$. The next stability result is a straightforward corollary of Theorem 4.4.3 and it is stated without proof.

Corollary 4.4.4. *The output dynamics of $y = \mathbf{M}_n x$ of (4.15) satisfy*

$$\dot{y} = -\mathcal{L}_d(y)\, y, \tag{$\mathcal{N}_d$}$$

with $y = 0$ is the a globally exponentially stable hyperbolic fixed point.

The dynamics of $(\mathcal{N}_d)$ satisfy $y(t, y_0) = \mathbf{M}_n\, S(t, x_0)$, $t \geq 0$. The energy of the output once weighted with a positive-definite and symmetric matrix $\mathbf{Q}$ is

$$\int_0^\infty y^T(t, y_0) Q\; y(t, y_0)\, dt.$$

We choose the performance measure as the mean energy of the vanishing signal y, when the state of the consensus system starts from a random initial condition $\mathbf{x}_0$. The long term energy of the output signal $\mathbf{y}$ that converges to zero is equivalent to the energy of the state vector x to converge to consensus. We take this mean for uncertain initial conditions, by assuming that the initial state is a random variable $x_0 : \Omega_s \to \Omega$ from the sample space Ω_s, with some probability measure (e.g. a probability density function or a probability mass function). In either case, we define the performance measure as

$$\rho\left(\mathcal{L}\right) := \mathbb{E}_{x_0}\left\{\int_0^\infty S^T(t,x_0)\mathbf{M}_n^T Q\ \mathbf{M}_n S(t,x_0)\,dt\right\}. \tag{4.18}$$

The next result establishes an analytical expression for the performance measure of $(\mathcal{N}_d)$ that reflects the contributions of the spectra of the linearized graph Laplacian and eigenfunctions of the Koopman operator.

Theorem 4.4.5. (Performance Measure) *Consider the disagreement dynamics* $(\mathcal{N}_d)$ *and the associated flow* $S(\cdot, y_0)$ *for all initial disagreements* y_0*. Then, the performance measure* (4.18) *can be expressed as*

$$\rho(\mathcal{L}) = \sum_{i,j\geq 1} \phi_{ij} c_{ij} \frac{1}{\bar{\lambda}_i + \bar{\lambda}_j}, \tag{4.19}$$

where $\left\{\bar{\lambda}_i\right\}_{i=1,2,\ldots}$ *is the sequence of Koopman eigenvalues in the KMD of* $(\mathcal{N}_d)$*, enumerated by an arbitrary numbering of* $\gamma_i = (j_2, \ldots, j_n) \in \mathbb{Z}_+^{n-1}$ *as*

$$\bar{\lambda}_i := \sum_{k=2}^{n} j_k \lambda_k \quad \textit{and} \quad \phi_i(x_0) := \prod_{k=2}^{n} H_k^{j_k}(x_0).$$

with $\lambda_2, \ldots, \lambda_n$ *being the nonzero eigenvalues of* $\mathcal{L}_0 := \frac{\partial}{\partial x}\mathcal{L}(x)|_{x=0}$*. Moreover,* $\phi_{ij} := \mathbb{E}_{x_0}\{\phi_i(y)\phi_j(y)\}$ *and* $c_{ij} := c_i^T Q\, c_j$*, are computed in terms of Koopman eigenfunctions and modes, respectively.*

Proof. The disagreement dynamics $(\mathcal{N}_d)$ attain a globally exponentially stable hyperbolic origin. In view of Assumptions 4.4.1 and 4.4.2, one can sort the eigenvalues of $-\mathbf{A} = \frac{\partial}{\partial y}\mathcal{L}_d(y)|_{y=0}$ as $\lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$ such that $\lambda_1 = \delta$. We claim that the restriction of

$\phi_1(x) = H_1(x)$ to $\mathbf{1}^\perp$ is zero, since $\phi_1(x) = \mathbf{1}_n^T x$. We substitute $\phi_1(x)$, $F(x) = -\mathcal{L}_d(x)x$, and $\lambda_1 = -\delta$ into the left hand side of (4.14) to obtain

$$\nabla^T \phi_1(x) F(x) = \mathbf{1}_n^T(-\mathcal{L}(x) - \delta \mathbf{J}_n/n)x,$$

which implies that

$$\nabla^T \phi_1(x) F(x) = 0 - \delta \sum_{i=1}^{n} x_i = -\delta \times \mathbf{1}_n^T x = -\lambda_1 \phi_1.$$

Therefore, $\phi_1(x) = \mathbf{1}_n^T x$ is in fact a Koopman eigenfunction with eigenvalue $-\delta$. We observe that for any $y \in \mathbf{1}^\perp$, it holds that $\phi_1(y) = 0$. Considering the restricted dynamics, $H_1(y) = \phi_1(y) = 0$. Hence, any Koopman eigenfunction parametrized with $\gamma_i = (j_1, j_2, \ldots, j_n)$ with $j_1 \geq 1$ is zero, because the corresponding eigenfunction is

$$\phi_i(x) = \prod_{k=1}^{n} H_k^{j_k}(x).$$

Now let a H^{-1} have the form (4.11) for y. We consider a KMD based on Proposition 4.3.5. This implies that all terms related to λ_1 are canceled out of the decomposition. Thus, we can restrict the numbering of summation indices to $\mathbb{Z}_+^{n-1}$ and then write the KMD for $y(\cdot, y_0) = \mathbf{M}_n S(\cdot, x_0)$ as

$$y(t, y_0) = \sum_{i \geq 1} c_i \mathrm{e}^{-\bar{\lambda}_i t} \phi_i(y_0) \tag{4.20}$$

where for any multi-index $\gamma_i = (j_2, \ldots, j_n) \in \mathbb{Z}_+^{n-1}$ inducing

$$\bar{\lambda}_i := \sum_{k=2}^{n} j_k \lambda_k \quad \text{and} \quad \phi_i(x_0) := \prod_{k=2}^{n} H_k^{j_k}(x_0).$$

The integrand of the integral in the performance measure is

$$y^T(t, y_0)\ Q\ y(t, y_0) = \left(\sum_{i \geq 1} \mathrm{e}^{-\bar{\lambda}_i t} \phi_i(y_0) c_i^T \right) \mathbf{Q} \left(\sum_{j \geq 1} c_j \mathrm{e}^{-\bar{\lambda}_j t} \phi_j(y_0) \right).$$

We reorganize this quadratic term as

$$y^T(t, y_0)\ \mathbf{Q}\ y(t, y_0) = \sum_{i,j\geq 1} \mathrm{e}^{-(\bar{\lambda}_i+\bar{\lambda}_j)t}\phi_i(y_0)\phi_j(y_0)c_i^T\mathbf{Q}\ c_j.$$

The induced eigenvalues satisfy $\bar{\lambda}_i = \sum_{k=2}^{n} j_k\lambda_k > 0$, hence, $\bar{\lambda}_i + \bar{\lambda}_j > 0$ for all $i, j \geq 1$. Integrating over all times yields

$$\int_0^\infty y^T(t, y_0)Q\ y(t, y_0)\ dt = \sum_{i,j\geq 1} \phi_i(y_0)\phi_j(y_0)\frac{c_i^T\mathbf{Q}c_j}{\bar{\lambda}_i + \bar{\lambda}_j}.$$

The result follows by virtue of the linearity of the expected value. □

4.4.1 Analytic Examples

The Koopman representation of flows in consensus networks can be derived analytically, for some special cases. In this section, we discuss a few such types of networks in the form of (4.15) where the associated Koopman modes (subsequently $\rho(\mathcal{L})$) can be calculated in a closed form.

Example 4.4.6 (Linear Consensus Network). We evaluate the performance measure of a first-order LTI consensus network of order n, which has the dynamics

$$\dot{x} = -\mathcal{L}\,x,$$

for a graph Laplacian $\mathcal{L}$ that is state-independent (i.e. $g \equiv 1$) but satisfies Assumption 4.4.2. To use (4.19), we let $\mathbf{Q} = \mathbf{I}_n$ and choose the initial conditions such that

$$\mathbb{E}_{x_0}\left\{y_0{y_0}^T\right\} = \mathbf{I}_n.$$

We denote the eigenvalues of $\mathcal{L}$ as λ_i for $i = 1, \ldots, n$. Based on Lemma 4.3.3, λ_i has a Koopman eigenfunction $\phi_i(x) = H_i(x)$, that is

$$\phi_i(x) = v_i^T x,$$

where v_i is the unit eigenvector of $\mathcal{L}$ corresponding to λ_i (see [105, 107]). Let $V = [v_1|v_2|\dots|v_n]$ be the orthonormal matrix of eigenvectors of $\mathcal{L}$, then for the disagreement dynamics we have $H(y) = \mathbf{V}^T y$. Since $(\mathbf{V}^T)^{-1} = (\mathbf{V}^{-1})^{-1} = \mathbf{V}$ the inverse of this map is $H^{-1}(y) = \mathbf{V}y$. This lets us compute the components of the performance measure as follows.

$$\phi_{ij} = \mathbb{E}_{x_0}\left\{\phi_i(y)\phi_j(y)\right\} = \mathbb{E}_{x_0}\left\{v_j^T y \cdot v_i^T y\right\},$$

for all $i, j = 2, \dots, n$. We rearrange to obtain

$$\phi_{ij} = \mathbb{E}_{x_0}\left\{v_j^T y y^T v_i\right\} = v_j^T \mathbb{E}_{x_0}\left\{yy^T\right\} v_i = v_j^T v_i = \delta_{ij},$$

since $\mathbb{E}_{x_0}\{yy^T\} = \mathbf{I}_n$ and $\mathbf{V}$ is orthonormal. Obviously, $H^{-1}(y) = \mathbf{V}y$ is a exact polynomial representation, thus

$$c_i = \begin{cases} v_i & \text{if } i = 2, \dots, n \\ 0 & \text{if } i = 1 \end{cases},$$

which allows to compute the coefficients used in the performance measure as

$$c_{ij} = c_i^T c_j = \begin{cases} \delta_{ij} & \text{if } i, j = 2, \dots, n \\ 0 & \text{if } i \text{ or } j = 1 \end{cases}.$$

We substitute these terms into the result in (4.19) to find

$$\rho(\mathcal{L}) = \sum_{i=2}^{n} \frac{1}{2\lambda_i}, \tag{4.21}$$

that is the $\mathcal{H}_2$-norm squared of a first order linear consensus network [26].

It turns out that for the case when we have only two agents, we may be able to compute the eigenfunctions analytically. The next two examples highlight this fact.

Example 4.4.7. Suppose that the network consists of two agents with dynamics dictated by

(4.15) and weight functions

$$w_{ij} = \frac{1}{(1 + (x_i - x_j)^2)^\alpha}, \tag{4.22}$$

for some constant $\alpha \in \mathbb{R}_+$. Parameter α in (4.22) defines how localized the interactions are within the network. As α increases, the agents update their states mainly with respect to their closest neighbors. In fact, the particular type of link implies that magnitude of interaction between two subsystems becomes weaker as their state becomes more different. For such a consensus network, in the case of two nodes, let p and q denote the states of the agents. Consequently, we can explain the interaction of these two nodes through the dynamics

$$\begin{bmatrix} \dot{p} \\ \dot{q} \end{bmatrix} = \frac{-1}{(1 + (p - q)^2)^\alpha} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix}.$$

Now, we turn into the disagreement dynamics $\mathcal{N}_d$ with $\delta = 1$[1], whose Jacobian at the origin attains the eigevnalues $\lambda_1 = -1$ and $\lambda_2 = -2$. Based on Lemma 4.3.3, each eigenvalue corresponds to a Koopman eigenfunction, say $\phi_1(x)$ and $\phi_2(x)$. We restrict the dynamics to $\mathbf{1}_n$, however, for the sake of simplicity, we denote the restricted variables with the same notation (i.e., p and q) . Hence, $[p, q]^T \in \mathbf{1}^\perp$; i.e., $p + q = 0$. We already know that

$$\phi_1(x) = \mathbf{1}_n^T x,$$

whose restriction to $\mathbf{1}^\perp$ is indeed zero. Once $\phi_2(x)$ is restricted to $\mathbf{1}^\perp$, we may compute it in an explicit fashion using (4.14), that is

$$\begin{bmatrix} \partial\phi_2/\partial p \\ \partial\phi_2/\partial q \end{bmatrix}^T \frac{-1}{(1 + (p - q)^2)^\alpha} \begin{bmatrix} q - p \\ p - q \end{bmatrix} = -2\phi_2.$$

[1]Note that choice of δ is arbitrary.

We let $z := p - q$ and use the chain rule to obtain

$$\frac{\partial \phi_2}{\partial p} = \frac{\partial \phi_2}{\partial z} \text{ and } \frac{\partial \phi_2}{\partial q} = -\frac{\partial \phi_2}{\partial z}.$$

Noting that the only free variable is now z, we can change the partial derivatives with respect to z. The resulting scalar ordinary differential equation is

$$\frac{2z}{(1+z^2)^\alpha}\frac{d\phi_2}{dz} = 2\phi_2,$$

which together with $\phi_2(0) = 0$ implies that

$$\phi_2 = \pm z \exp\left(\int \frac{(1+z^2)^\alpha - 1}{z} dz\right).$$

Without loss of generality, we choose to work with the plus sign. Using the binomial series we write the numerator of the integrand as

$$h(z) := (1+z^2)^\alpha - 1 = \alpha z^2 + \frac{(\alpha)(\alpha-1)z^4}{2!} + \ldots,$$

for all $|z| < 1$. We integrate the series to get

$$\int \frac{h(z)}{z} dz = \sum_{n=1}^{\infty} \frac{\alpha(\alpha-1)\ldots(\alpha-n+1)z^{2n}}{2n \times n!}$$

which completes the evaluation of $\phi_2(z)$ as

$$\phi_2(z) = z \exp\left(\sum_{n=1}^{\infty} \frac{\alpha(\alpha-1)\ldots(\alpha-n+1)z^{2n}}{2n \times n!}\right),$$

that is a convergent series for $|z| < 1$, since exp(.) is analytic everywhere. The identity $p + q = 0$ implies $z = 2p$, thus

$$\phi_2(p) = 2p \exp\left(\sum_{n=1}^{\infty} \frac{\alpha(\alpha-1)\ldots(\alpha-n+1)2^{2n-1}p^{2n}}{n \times n!}\right).$$

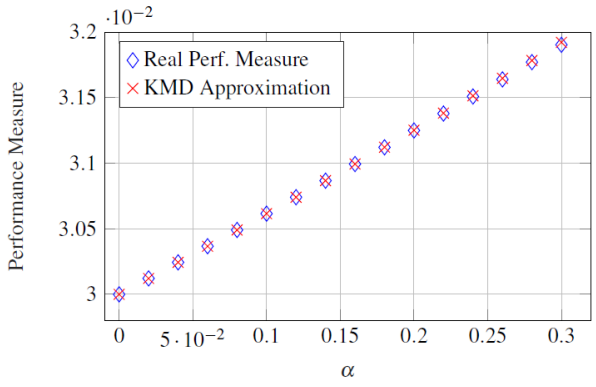

Figure 4.1: The performance measure of the network of two agents in Example 4.4.7 with the decaying parameter α.

Thus, the component-wise description of $H(p, q)$ is

$$H(p, q) = H(p) = \begin{bmatrix} \phi_2(p) & 0 \end{bmatrix}^T.$$

The definition of inverse of a map implies

$$H^{-1}(H(p, q)) = \begin{bmatrix} p & q \end{bmatrix}^T$$

so $H_1^{-1}(p, q)$ and $H_2^{-1}(p, q)$ are simply the inverse functions of $\phi_2(p)$ and $-\phi_2(p)$, respecively. These functions are locally analytic around the origin based on Lagrange inversion theorem [114]. Futhermore, one can calculate their coefficients in terms of Bell polynomials [115]. We assume that the initial value of p to come from a discrete random variable that takes the values from $\{0, 0.1, 0, 2, 0.3, 0.4\}$ with the uniform probability distribution. Because the mean of each initial condition must be zero, the initial value of q will be $-p$. We compute the value of $\rho(\mathcal{L})$ for $\mathbf{Q} = \mathbf{I}_n$ using the 17 order Maclaurin series of both eigenfunctions and the inverse map $H^{-1}(p)$. Because the value of the performance measure may be computed from the numerical integration of the trajectory as well, we may compare the results of the analytic approximations to the KMD and the real value of performance measure. These two value for a range of $\alpha \in [0, 0.4]$ have been illustrated in Fig. (4.1, where they are in good numerical agreement. The relative error is observed to increase from zero in the case that $\alpha = 0$ to less than 0.13% for $\alpha = 0.4$.

Remark 14. There is a limitation on the magnitude of the admissible initial conditions for the analysis conducted in the previous example. However, notice that this does not imply that it is a linear analysis since for any value of α, the linearization matrix of $\mathcal{N}_d$ (with $k = 1$) in Example 4.4.7 is the following matrix.

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} + \mathbf{J}_n.$$

This means that the value of the performance measure computed using the linearized system for each value of decaying parameter α is only one value.

Example 4.4.8. The dynamics of oscillators have been observed to be closely related to the consensus dynamics. Kuramoto suggested a model of biological oscillation, in which each oscillator was connected to the other one; i.e., the topology was a complete graph. Instead the interactions can be limited over a certain graph [89], so the dynamics of agent i can be represented as

$$\dot{x}_i = \omega_i + \sum_{\{i,j\}\in\mathcal{E}} w_{ij}(x_j - x_i), \tag{4.23}$$

where ω_i is the natural frequency and the coupling weight is

$$w_{ij} = K\frac{\sin(x_i - x_j)}{x_i - x_j}. \tag{4.24}$$

When the agents are identical (i.e., when $\omega_i = \omega$ for some ω), the change of variable $x_i \to x_i - \omega t$ induces a nonlinear consensus network that is

$$\dot{x}_i = \sum_{\{i,j\}\in\mathcal{E}} w_{ij}(x_j - x_i). \tag{4.25}$$

We proceed with a procedure for computation of performance measure similar to one introduced in Example 4.4.7. For two identical oscillators, with phases θ and γ, the dynamics

are

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \gamma\end{bmatrix} = K\begin{bmatrix}\sin(\gamma-\theta)\\ \sin(\theta-\gamma)\end{bmatrix},$$

Setting $k = 1$, the disagreement Jacobian has eigenvalues $\lambda_1 = -1$ and $\lambda_2 = -2K$, with eigenfunctions ϕ_1 and ϕ_2, respectively. Again we only need the restriction of ϕ_2 to $\mathbf{1}^\perp$. The equation (4.14) for these dynamics becomes

$$\begin{bmatrix}\partial\phi_2/\partial\theta\\ \partial\phi_2/\partial\gamma\end{bmatrix}^T K\sin(\theta-\gamma)\begin{bmatrix}1\\ -1\end{bmatrix} = -2K\phi_2.$$

The new variable $z := \theta - \gamma$ creates a single ordinary differential equation that is

$$\frac{2\sin(z)}{d\phi_2}dz = 2\phi_2 \Rightarrow \frac{d\phi_2}{\phi_2} = \frac{1}{\sin(z)}dz,$$

which is integrated and manipulated to get

$$\phi_2 = \pm\exp\left(-\ln(\cot(z/2))\right) = \pm\tan(z/2),$$

where we arbitrarily choose $+$. The eigenfunction ϕ_2 is locally analytic around the origin for $|z| < \pi$. Restricting to $\mathbf{1}^\perp$, $\theta + \gamma = 0$, so $z = 2\theta$, hence

$$\phi_2(\theta,\gamma) = \phi_2(\theta) = \tan(\theta),$$

which helps write the components of $H(\theta,\gamma)$ as $H(\theta,\gamma) = \begin{bmatrix}\phi_2 & 0\end{bmatrix}^T$. First component of $H^{-1}(\theta)$ satisfies $H_1^{-1}(\phi_2(\theta),0) = \theta$. Hence, $H_1^{-1}(\theta,\gamma) = \arctan(\theta)$, which is again locally analytic for $|\theta| < 1$ around zero.

We sample initial conditions from a uniform discrete random variable of 63 equally distributed initial conditions in $\theta \in [0,\pi/5]$ (and $\gamma = -\theta$) with equal distance of 0.01. We set $\mathbf{Q} = \mathbf{I}_n$ and use 17^{th} order Maclaurin series of eigenfunctions and $H^{-1}(\theta)$ to assess $\rho(\mathcal{L})$. As shown in Fig. 4.2, we alter $K \in [0.2,4]$ and take a look at the values of the

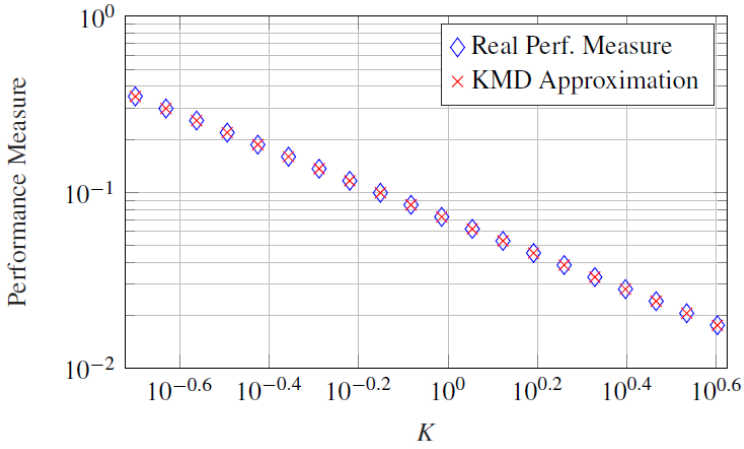

Figure 4.2: The performance measure of the Kuramoto model of two agents in Example 4.4.8 with the parameter K.

performance measure, once compared with the exact value of $\rho(\mathcal{L})$ (computed with the numerical integration of the trajectories). The numerical agreement in this experiment can be measured by the relative error of the performance using the KMD approximation, which was about $3 \times 10^{-4}\%$ in the worst-case.

4.5 Sparse Polynomial Approximations

We have observed that any polynomial approximation to the inverse map of eigenfunctions $H^{-1}(x)$ will result in the Koopman Mode Decompositions. In this section, first, we detail a general sparse approximation technique for multivariate interpolation. Then, we demonstrate how we can use this technique for the map of Koopman eigenfunctions $H(x)$ as well as the inverse $H^{-1}(x)$.

4.5.1 Smolyak-Collocation Method

To introduce the notion of sparsity for the approximation of both Koopman eigenfunctions and Koopman Mode Decomposition, we may use sparse functional approximation methods. The idea is that instead of searching for the approximant in the whole space of polynomials, the search is carried out over a nearly optimal sparse basis, called *Smolyak basis*. The output of the method would be a polynomial: the weighted sum of the tensor product of Chebyshev

polynomials that are in the basis. Naturally, we choose the coefficients of the polynomial with respect to some error criterion. One way of doing so is collocation, where we enforce the approximant to (perhaps approximately) satisfy the governing equation of the problem at the given points of a grid, called Smolyak Sparse Grid. To describe this method, we need few basic tools (consult [116] and [117] for more details).

Definition 4.5.1 (Chebyshev Polynomials)**.** *The sequence of the (scalar) Chebyshev polynomials of first kind* $\{T_i(x)\}_{i=1,2,\dots}$ *are initialized with* $T_1(x) = 1$ *and* $T_2(x) = x$ *and recursively defined as follows.*

$$T_{i+1}(x) = 2xT_i(x) - T_{i-1}(x), \text{ for } i = 2, 3, \dots \tag{4.26}$$

Similarly, the Chebyshev polynomials of second kind $\{U_i(x)\}_{i=1,2,\dots}$ *start with* $U_1(x) = 1$ *and* $U_2(x) = 2x$*, and then iteratively*

$$U_{i+1}(x) = 2xU_i(x) - U_{i-1}(x), \text{ for } i = 2, 3, \dots \tag{4.27}$$

We define the integer function $m(i) : \mathbb{N} \to \mathbb{N}$ with $m(1) = 1$ and for $i = 2, 3, \dots$, it is evaluated according to

$$m(i) := 2^{i-1} + 1. \tag{4.28}$$

We also define the sequence of sets $\{\mathcal{G}^i\}_{i=1,2,\dots}$ wherein, $\mathcal{G}^1 = \{0\}$, and for $i = 2, 3, \dots$, it holds that $\mathcal{G}^i = \{\zeta_1, \dots, \zeta_i\} \subset [-1, 1]$, that is the set of the extrema of the Chebyshev polynomials with the components given by

$$\zeta_j := -\cos\left(\frac{\pi(j-1)}{i-1}\right), \text{ for all } j \in \{1, \dots, i\}. \tag{4.29}$$

In the next definition, we use the multi-index notation $\mathbf{i} = (i_1, \dots, i_n) \in \mathbb{N}^n$ inducing $|\mathbf{i}| := \sum_{k=1}^{n} i_k$.

Definition 4.5.2 (Smolyak Sparse Grid)**.** *The Smolyak Sparse grid of* $[-1, 1]^n$ *is a union*

of the Cartesian products of the form

$$\mathcal{H}^{n,\mu} := \bigcup_{|\mathbf{i}|=n+\mu} \left(\mathcal{G}^{m(i_1)} \times \cdots \times \mathcal{G}^{m(i_n)} \right), \tag{4.30}$$

where positive integer $\mu \in \mathbb{N}$ *is the order of the grid*[2].

Definition 4.5.3 (Smolyak Approximant Polynomial)**.** *The Smolyak approximant to a function* $f : [-1,1]^n \to \mathbb{R}$ *is given by*

$$\hat{f}^{n,\mu}(x) := \sum_{q \le |\mathbf{i}| \le n+\mu} (-1)^{n+\mu-|\mathbf{i}|} \binom{n-1}{n+\mu-|\mathbf{i}|} \mathrm{p}^{\mathbf{i}}(x), \tag{4.31}$$

with $q = \max(n, \mu+1)$ *the tensor product polynomials for each multi-index* $\mathbf{i} = (i_1, \ldots, i_n)$ *defined as*

$$\mathrm{p}^{\mathbf{i}}(x) := \sum_{l_1=1}^{m(i_1)} \cdots \sum_{l_n=1}^{m(i_n)} \theta_{l_1,\ldots,l_n} T_{l_1}(x_1) \ldots T_{l_n}(x_n), \tag{4.32}$$

where $\theta_{l_1,\ldots,l_n}$ *the coefficients that are to be determined.*

To find the optimal vector of coefficients of $\Theta = \mathrm{vec}(\theta_{l_1,\ldots,l_n}) \in \mathbb{R}^m$ for approximation of some function f that is $C^k([-1,1]^n)$, an error objective should be defined and minimized. One way is to consider the error function $E(f,\Theta) : C^k([-1,1]^n) \times \mathbb{R}^n \to \mathbb{R}_+$ to be

$$\begin{aligned} E(f,\Theta) &:= \int_{[-1,1]^n} |f(x) - f^{n,\mu}(x)|^2 \prod_{y \in \mathcal{H}^{n,\mu}} \delta(x-y)\, \mathrm{d}x \\ &= \sum_{y \in \mathcal{H}^{n,\mu}} |f(y) - f^{n,\mu}(y)|^2 , \end{aligned}$$

where δ is the Dirac's delta function. This metric is certainly minimized (i.e., $E(f,\Theta) = 0$) if Θ is chosen such that

$$f(x) = f^{n,\mu}(x) \text{ for all } x \in \mathcal{H}^{n,\mu}. \tag{4.33}$$

This procedure is called collocation. A pivotal property of the overall method is that size

[2]One can show that grids of higher order include all grids of lower order; i.e., $\mathcal{H}^{n,\mu} \subset \mathcal{H}^{n,\mu+1}$.

of vector of coefficients Θ and the number of interpolation points is equal; i.e., $|\mathcal{H}^{n,\mu}| = |\Theta| = M$. Therefore, enforcing equalities (4.33) constitutes of searching for the solution to M equations involving M unknown entries of Θ. Once we evaluate the coefficients with the described scheme, the following error bound would hold, wherein the used functional norm $\|.\| : C^k\left([-1,1]^n\right) \to \mathbb{R}_+$ is defined as

$$\|f\| = \max\left\{\left\|D^i f\right\|_\infty : \ i = 1, \ldots, k\right\}. \tag{4.34}$$

Theorem 4.5.4 (Theorem 2 in [116]). *Suppose that function $f(x) : [-1,1]^n \to \mathbb{R}$ is $C^k\left([-1,1]^n\right)$, together with a Smolyak approximant $\hat{f}^{n,\mu}(x)$ that interpolates f on $\mathcal{H}^{n,\mu}$ with $|\mathcal{H}^{n,\mu}| = M$. Then, for some positive constant $c_{n,k}$, the error of the approximation is bounded according to*

$$\left\|f - \hat{f}^{n,\mu}\right\| \leq c_{n,k} M^{-k} (\log M)^{(k+2)(n+1)+1}. \tag{4.35}$$

Each multi-index $\mathbf{i}$ in (4.31) induces a number of tensor product polynomials that are summed together as in (4.32). We gather the indices of all these tensor product polynomials in a set $\mathbf{L}^{n,\mu}$; i.e.,

$$\mathbf{L}^{n,\mu} := \bigcup_{q \leq |\mathbf{i}| \leq n+\mu} \{(l_1, \ldots, l_n) : l_j \leq m(i_j)\}. \tag{4.36}$$

One can show that at the end of the day, the approximant constructed in (4.31) using the polynomials (4.32) boils down to the following simple representation

$$\hat{f}^{n,\mu}(x) = \sum_{\mathbf{l}_i = (l_1^i, \ldots, l_n^i) \in \mathbf{L}^{n,\mu}} \Theta_i \mathrm{T}_i(x) = \sum_{i=1}^{M} \Theta_i \mathrm{T}_i(x), \tag{4.37}$$

where $M = |\mathcal{H}^{n,\mu}| = |\mathbf{L}^{n,\mu}|$, and for $\mathbf{l}_i = (l_1^i, \ldots, l_n^i) \in \mathbf{L}^{n,\mu}$, the coefficients Θ_i and polyno-

mial terms are given by

$$\Theta_i := \theta_{l_1^i,\ldots,l_n^i}, \tag{4.38}$$

$$\mathrm{T}_i(x) := T_{l_1^i}(x_1)\ldots T_{l_n^i}(x_n). \tag{4.39}$$

To compute the partial derivatives of approximation, we use the definitions of the Chebyshev polynomials to define

$$\mathrm{T}_i^j(x) := \begin{cases} \mathrm{T}_i \cdot \dfrac{U_{l_j^i}(x_j)}{T_{l_j^i}(x_j)} & l_j^i = 2,\ldots,n \\ 0 & l_j^i = 1 \end{cases}, \tag{4.40}$$

This lets us write the partial derivatives of $\hat{f}^{n,\mu}$ in the compact form

$$\frac{\partial \hat{f}^{n,\mu}(x)}{\partial x_j} = \sum_{i=1}^{M} l_j^i \Theta_i \mathrm{T}_i^j(x). \tag{4.41}$$

4.5.2 Sparse Approximation to Eigenfunctions

We denote the approximation to Koopman eigenfunction $\phi(x)$ by $\hat{\phi}(x)$. Substituting (4.37) and (4.41) into (4.14), we get the (approximate) equality

$$\sum_{j=1}^{n}\sum_{i=1}^{M} l_j^i \Theta_i \mathrm{T}_i^j(x) F_j(x) \approx \lambda \sum_{i=1} M \Theta_i \mathrm{T}_i(x).$$

We change the order of summations to further obtain

$$\sum_{i=1}^{M}\left(\sum_{j=1}^{n}\left(l_j^i \mathrm{T}_i^j(x) F_j(x)\right) - \lambda \mathrm{T}_i(x)\right)\Theta_i \approx 0. \tag{4.42}$$

We define and denote the vector of coefficients $\Theta \in \mathbb{R}^M$ by $\Theta := [\Theta_1,\ldots,\Theta_M]^T$. For a point in the grid $x^k \in \mathcal{H}^{n,\mu}$, we may write the left hand side of (4.42) as

$$\mathcal{A}_k\Theta = [\mathcal{A}_{ki}]_{i=1,\ldots,n}\,\Theta,$$

where the entries of row vector $\mathcal{A}_k \in \mathbb{R}^{1\times M}$ can be computed from

$$\mathcal{A}_{ki} := \sum_{j=1}^{n} \Big(l_j^i \mathrm{T}_i^j(x^k) F_j(x^k) \Big) - \lambda \mathrm{T}_i(x^k), \quad \text{for all } i = 1, \ldots, M.$$

Repeating this for M points in the Smolyak grid, the stacked left hand side of all equations becomes $\mathcal{A}\Theta$, where $\mathcal{A} \in \mathbb{R}^{M\times M}$ is given by

$$\mathcal{A} := [\mathcal{A}_1^T, \ldots, \mathcal{A}_M^T]^T. \tag{4.43}$$

Ideally, $\mathcal{A}\Theta$ should be zero for an eigenfunction, however, if it is not possible, we would like to minimize an error function, which we choose to be

$$J(\Theta) = \|\mathcal{A}\Theta\|_2^2. \tag{4.44}$$

Now, because $\mathbf{A}$ may have repeated eigenvalues, we add a constraint that lets us derive multiple eigenfunctions corresponding to one eigenvalue. Consider the Koopman eigenfunction $\phi(x)$ with Koopman eigenvalue λ, which is the eigenvalue of the linearization matrix with a left eigenvector $\mathbf{R} = [r_1, \ldots, r_n] \in \mathbb{R}^{n\times n}$. We omit the index of the eigenvalues and eigenvectors in the following developments for simplicity and consider λ to be associated with the eigenvector $r \in \mathbb{R}^n$. We can show that with the fixed point at the origin,

$$\nabla \phi(x)|_{x=0} = r.$$

Translating this for the approximant, for each $j = 1, \ldots, n$ we have

$$\frac{\partial \hat{\phi}(x)}{\partial x_j}|_{x=0} = \sum_{i=1}^{M} l_j^i \Theta_i \mathrm{T}_i^j(0) = \mathcal{B}_j \Theta = r_j,$$

where the row vector $\mathcal{B}_j \in \mathbb{R}^{1\times M}$ has the components

$$\mathcal{B}_{ji} = l_j^i \mathrm{T}_i^j(0), \quad \text{for all } i = 1, \ldots, M.$$

The matrix form of this equality becomes

$$\mathcal{B}\Theta = r, \tag{4.45}$$

where the matrix $\mathcal{B} \in \mathbb{R}^{n \times M}$ is the result of stacking row vectors as

$$\mathcal{B} = \left[\mathcal{B}_1^T, \ldots, \mathcal{B}_n^T\right]^T. \tag{4.46}$$

Recall that our approximation requires $\hat{\phi}(0) = 0$ to provide exponential convergence for the performance measure integrals (see Remark 13). This condition can be translated to single scalar equality

$$\mathcal{C}\Theta = 0, \tag{4.47}$$

where $\mathcal{C} \in \mathbb{R}^{1 \times M}$ is the row vector with elements

$$\mathcal{C}_i := \mathrm{T}_i(0) \quad \text{for all } i = 1, \ldots, M.$$

Now, we would like to minimize the error function defined by (4.44), while constraints (4.45) and (4.47) are satisfied. We define the optimization problem

$$\begin{aligned} &\underset{\Theta \in \mathbb{R}^m}{\text{minimize}} \quad \|\mathcal{A}\Theta\|_2^2, \\ &\text{subject to} \quad \begin{bmatrix} \mathcal{B} \\ \mathcal{C} \end{bmatrix} \Theta = \begin{bmatrix} r \\ 0 \end{bmatrix}. \end{aligned} \tag{4.48}$$

This program is equivalent to a Semi-Definite Program (SDP) and can be solved using the conventional convex programming such as CVX [39].

4.5.3 Sparse Approximation to Koopman Mode Decomposition

In the previous subsection, we illustrated a way to find approximations to the Koopman eigenfunctions. Thus, the components of the map $H(x)$ can be approximated. Here, fol-

lowing a similar approach, we seek approximations to the components of its inverse map $H^{-1}(x)$. The very natural equation for component $H_j^{-1}(x)$ is

$$H_j^{-1}(H(x)) = x_j, \quad \text{for all } j = 1, \ldots, n. \tag{4.49}$$

We only have an approximation to $H(x)$, namely $\hat{H}(x)$, and we need approximations to $\hat{H}^{-1}(x)$, namely $\hat{H}^{-1}(x)$. Hence, we consider the approximate equality

$$\hat{H}_j^{-1}(\hat{H}(x)) \approx x_j, \quad \text{for all } j = 1, \ldots, n. \tag{4.50}$$

Again, following the spirit of the collocation method, for each point of the grid $x^k \in \mathcal{H}^{n,\mu}$, we enforce this equation to hold. Suppose that we have found the series approximation to each components of $H(x)$, denoted by $\hat{H}(x)$. Moreover, we define

$$z^k := \hat{H}(x^k). \tag{4.51}$$

Then, we consider a Smolyak series representation for this function as

$$\hat{g}_j^{n,\mu}(z^k) = \sum_{i=1}^{M} \Phi_i^j \mathrm{T}_i(z^k). \tag{4.52}$$

Similar to essence of the method that we discussed in the previous subsection, we define vector of coefficients $\Phi_1^j \in \mathbb{R}^M$ to be

$$\Phi^j := \left[\Phi_1^j, \ldots, \Phi_M^j\right].$$

Inserting (4.52) into (4.49), we get

$$\mathcal{D}_k \Phi^j = \left[\mathcal{D}_{ki}\right]_{i=1,\ldots,M} \Phi^j \approx x_j^k, \tag{4.53}$$

where the components of row vector $\mathcal{D}_k \in \mathbb{R}^{1\times M}$ are

$$\mathcal{D}_{ki} := \mathrm{T}_i(z^k). \tag{4.54}$$

Concatenating these vectors and the right hand side scalars for each point in the grid (i.e., M points), we may write these equations as

$$\mathcal{D}\Phi^j \approx X_j, \tag{4.55}$$

where matrix $\mathcal{D} \in \mathbb{R}^{M\times M}$ and vector $X_j \in \mathbb{R}^M$ are given by

$$\mathcal{D} := \left[\mathcal{D}_1^T, \ldots, \mathcal{D}_M^T\right]^T, \tag{4.56}$$

$$X_j := \left[x_j^1, \ldots, x_j^M\right]^T, \tag{4.57}$$

respectively. Again one hopes that (4.55) holds with a minimal error for each $j = 1, \ldots, n$. Therefore, we define the optimization problem

$$\underset{\Phi^j \in \mathbb{R}^M}{\text{minimize}} \quad \left\|\mathcal{D}\Phi^j - X_j\right\|_2^2. \tag{4.58}$$

Note that matrix $\mathcal{D}$ is deliberately denoted without index j, because it is the same matrix for the optimization problem for each component $H_j^{-1}(x)$. The solution to this least-squares optimization problem is given by

$$\Phi^j = \mathcal{D}^\dagger X_j \quad \text{for all } j = 1, \ldots, n.$$

We should repeat this for each component of $H^{-1}(x)$. Putting the results in a matrix gives us

$$\Phi := \left[\Phi^1, \ldots, \Phi^n\right].$$

Because the value of matrix $\mathcal{D}$ is shared between M optimization problems defined by (4.58), by inspection, we find that

$$\Phi = \mathcal{D}^\dagger \mathbf{X}^T, \tag{4.59}$$

where $\mathbf{X} \in \mathbb{R}^{n\times M}$ is the matrix containing the vector of all grid points. Now, we have a polynomial approximation to H^{-1}, which would give us a Koopman Mode Decomposition.

4.5.4 Numerical Examples

In this section, we show that we may be able to effectively estimate the performance measure of nonlinear consensus networks with more than two subsystems. Note that in all cases, the real performance measure is calculated from the numerical solution of the network output followed by numerical integration.

Example 4.5.5 (Complete Graphs). Using the described numerical approximation method, we estimate the performance measure for the nonlinear consensus network with exponentially decaying weights defined in (4.22). The corresponding linearized graph Laplacian corresponds to the undirected unweighted complete graph; i.e.

$$\mathbf{A} = -\mathcal{L}_{\mathcal{K}_n} = \mathbf{J}_n/n - \mathbf{I}_n.$$

We evaluate the performance measure from the KMD approximation based on the numerical integration of the solutions. The initial conditions are uniformly sampled random initial conditions from $[-1, 1]^n$. The numerical values for data using Koopman approach have been obtained by implementation of the suggested numerical method with and the results are shown in Fig. 4.3. In this example, the error in the evaluated performance measure using our numerical method is less than 2%.

Example 4.5.6 (Random Graphs). We fix $N = 8$ and create Erdős-Rényi graphs with different edges probabilities (and consequently, different edge numbers). Then, we consider again the exponentially decaying weights given by (4.22). The performance measures from Monte-Carlo simulations as well as the formula (using the method discussed in the previous sections) are also evaluated and illustrated in Fig. 4.4. The error of approximation, in this case, is less than 1.4%.

4.5.5 Comparison to Extended Dynamic Mode Decomposition

The numerical method explained in this section is related to the notion of Extended Dynamic Mode Decomposition (EDMD) [110], which has been a promising procedure for extracting information about the Koopman spectrum of the dynamical system. In EDMD, to find a

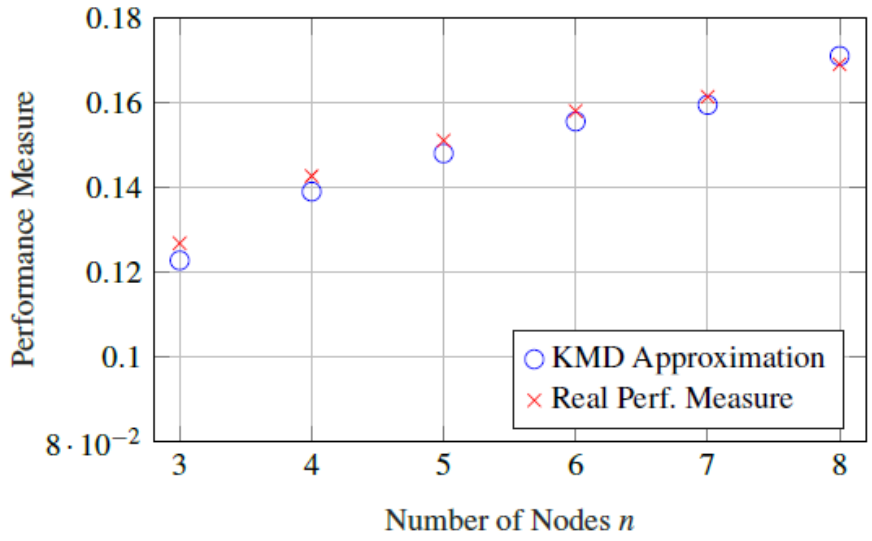

Figure 4.3: The performance measure of nonlinear consensus network with $\alpha = 0.25$ and the graph at the linearized Laplacian of complete graph.

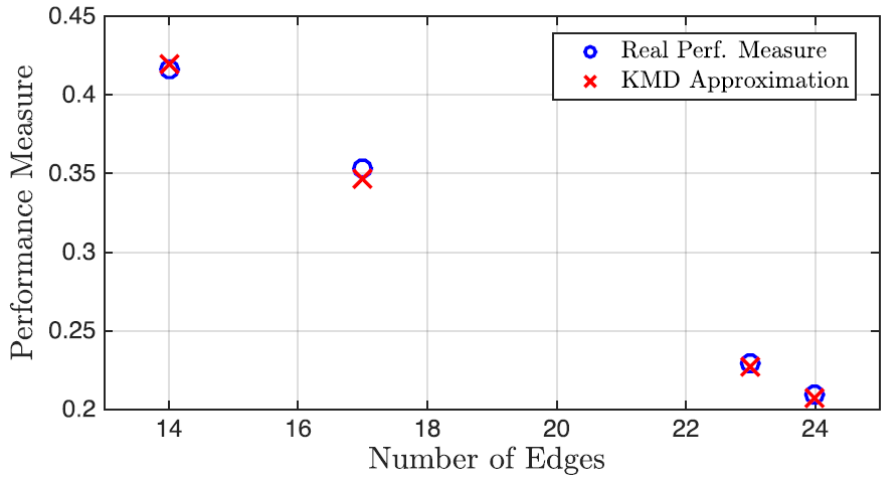

Figure 4.4: The performance measure of nonlinear consensus network with $N = 8$, $\alpha = 0.25$ and random graphs with different number of edges

KMD for the flow of the dynamical system, one should first assume a rich enough dictionary of basis functions such that hopefully, the Koopman eigenfunctions lie in their span. Then, using the snapshots from the trajectory, one may find a truncated approximation to the Koopman operator and finite number of approximations to the Koopman eigenfunctions and their corresponding eigenvalues. Then, the solution to the dynamical system is approximated as a truncated KMD using those eigenvalues and eigenfunctions.

In the current settings, we know what are the eigenvalues and eigenfunctions that are required for representation of the flow of the nonlinear system. Hence, we do not need the first step of the EDMD for the computation of the (approximate) eigenvalues and eigenfunctions. In fact, we build the dictionary that one needs for EDMD based on the principal eigenfunctions in the map $H(x)$.

On the other hand, the second step in both methods are connected in the spirit: In our approach, we find an approximation to map $H^{-1}(x)$ using identity

$$H^{-1}\left(H(x)\right) = x.$$

While in EDMD, the identity observable (i.e., $\mathbf{f}(x) \equiv x$) has to be represented in terms of the basis functions in the dictionary. Then one is allowed to write down a KMD for the system dynamics as explained before.

4.6 Conclusion and Discussion

Koopman mode decomposition approaches hold promise for performance analysis and synthesis of nonlinear dynamical systems, that are of interest in various disciplines of engineering and control. The vital connection between the eigenspectrum of linearized dynamics and Koopman operator provides a closed form evaluation of the first moment of energy integral of the solutions, in terms of Koopman eigenvalues, eigenfunctions, and modes. The numerical approximation of KMD components is implemented by a scalable computational algorithm using sparse Smolyak grid with certifiable accuracy. Future directions include, but are not limited to the following directions: investigation and analysis of various performance metrics in nonlinear systems as extensions of linear control systems [26]. Another research line regards dynamical systems with higher order integrators as well as a systemic performance-based network synthesis for optimal interactions among interconnected entities in the face of uncertain initial conditions or other structural parameters.

Part II:

Estimation Problems in Complex Systems

Chapter 5

Space-Time Sampling for Network Observability

5.1 Introduction

A common assumption in classical control systems for state estimation is that samples are collected periodically from some prescribed output sensors [118–120]. In practice, sampling strategies are designed subject to some given performance criteria and hardware/software constraints, e.g., achieving certain estimation quality, data processing power, battery-life of the sensors and processors, etc. Although over-sampling may result in superior estimation quality, it is usually undesirable in networked systems that are equipped with spatially distributed sensors; examples include, spatially distributed networked robots, synchronous power networks, and platoon of self-driving vehicles. In these applications, designing sparse sampling strategies, that allow collecting samples aperiodically from only a fraction of subsystems, will reduce sensing costs due to the existing algorithmic, physical, hardware, and software constraints. These burdens are even more pronounced in networks with several thousands subsystems. Our goal in this chapter is to propose a formal method to study properties and performance of various sampling strategies and devise scalable algorithms to design sparse sampling strategies in space and time with provable performance bounds.

There have been recent interest on revisiting notion of observability in the context of

networked control systems. In [121], the author revisit the notion of observability radius for a class of linear networks whose state matrices are adjacency matrix of some weighted graphs. They provide conditions to verify whether such networks can preserve observability property in presence of structured (weighted) edge perturbations. The authors also suggest a heuristic algorithm to compute size of perturbations that result in loss of observability by finding their smallest Frobenius norm. In [122] and [123], the problem of minimum constraint input selection are considered, where the objective is to find the smallest subset of inputs to ensure controllability. While it is shown that in general this problem is NP-hard, a subclass of such problems (by assuming dedicated inputs) can be solved efficiently with the aid of network graph algorithms. The author of [124] shows that the problem of approximating the minimum number of input (output) variables to guarantee controllability (observability) is NP-hard. It is shown that one can find an efficient approximation of the problem by employing a greedy heuristic to select variables to maximize the rank increase of the controllability matrix. We refer to [125] for some related background and earlier works in this context. In [126], the authors propose the problem of sensing-constrained LQG control, where contrary to the classical LQG, they look at the minimal sensing requirements for a desired control objective and tackle the problem by solving for suboptimal sensing strategies with by focusing only on finitely many sensing schedules. They have also looked at batch sensor-scheduling for linear-time invariant control systems [127], where the goal is to design a sensing strategy to near-optimally estimate the concatenated states in a given finite horizon. In [128], classic Kalman filtering problems were extended to the case with intermittent observations. Our work is close in spirit to [129, 130], where the authors investigate controllability of linear-time invariant networks and utilize randomized algorithms for sparsification [51] and (greedy) deterministic algorithms to obtain a sparse actuator scheduling. Furthermore, by allowing scaling in control inputs, they show that one can achieve desired levels of performance with respect to a class of performance measures. Prior to their work, several authors had also considered problems related to sensor or actuator scheduling for control and estimation, for instance see [131–134] and the references therein. In another related work [135], the authors look at the sensor placement problem for optimal parameter estimation and provide a near-optimal greedy algorithm for sensor

selection.

In this chapter, our focus is on estimating the initial condition of a linear time-invariant (LTI) network from a set of state samples that are collected sparsely from a subset of subsystems aperiodically over some time interval. This problem is closely related to state observer design with sensing-constraints for linear dynamical networks. In Section 6.3, we apply tools from (finite) frame theory to reformulate the network observability problem and show that one can extract an observability frame from any given set of samples that solves the observability problem. This key idea allows us to cast observability condition as whether a set of vectors forms a frame for the Euclidean space. This is particularly useful as every frame element is labeled by *where* and *when* it was taken. In Section 5.5, two types of measures, namely, standard deviation and differential entropy of the estimation error, are utilized to quantify quality of estimation for a given observability frame. They are also useful when one desires to compare estimation quality of various sampling strategies for a given network. We show that these estimation measures can be quantified using eigenvalues of the corresponding frame matrix. An important property of these estimation measures is that they are monotone with respect to the number of samples: by increasing the number of samples, the estimation measure does not deteriorate. In Section 5.6, we propose deterministic and randomized methods to generate observability frames for a given LTI network. It is shown that minimum required number of samples from each subsystem (location) depends on the degree of the minimal polynomial of the state matrix. We show in Section 5.8 that there are inherent fundamental limits on the best achievable levels of estimation quality, and intrinsic tradeoffs reveal an interplay between space-time samples: taking less samples (in average) per subsystem mandates collecting samples from more subsystems. In Section 5.7, we discuss three methods for frame sparsification: (i) sparsification by leverage scores, which is developed based on notions of spectral graph sparsification [51], (ii) random partitioning using Kadison-Singer paving solution [136], and (iii) greedy elimination using Sherman-Morrison rank-one update rule [137]. In all these algorithms, we obtain explicit error bounds for estimation-quality loss. We assert that our bounds are rather conservative. The reason is that, contrary to the results of [1, 51, 129, 130], elements of a sparsified observability frame cannot be rescaled to compensate for estimation-quality loss. At the

end, we support our theoretical finding by several simulation case studies.

5.2 Notations

The set of complex numbers, real numbers, nonnegative numbers, integers and nonnegative integers are shown by $\mathbb{C}$, $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$, respectively, and the imaginary number $\sqrt{-1}$ by j. For a given number $\gamma \in \mathbb{C}$, we define $\gamma\mathbb{Z} := \{\gamma k \mid k \in \mathbb{Z}\}$. For the n-dimensional Euclidean space $\mathbb{R}^n$, we denote its standard basis by $\{e_1, \ldots, e_n\}$ and the inner product of $x, y \in \mathbb{R}^n$ by $\langle x, y\rangle$. For a vector $x \in \mathbb{R}^n$, $\|x\|$ stands for its Euclidean 2-norm. For two families of vectors Φ_1 and Φ_2, $\Phi_1 \subset \Phi_2$ implies that $\phi \in \Phi_2$ for all $\phi \in \Phi_1$. We use the block capital letters to denote a matrix or a linear operator, e.g., $\mathbf{X}$. The transpose of a matrix $\mathbf{X}$ is denoted by $\mathbf{X}^T$, the matrix exponential of a square matrix $\mathbf{X}$ by $\mathrm{e}^{\mathbf{X}}$, and the identity matrix of appropriate size by $\mathbf{I}$. Eigenvalues of a positive semi-definite matrix $\mathbf{X} \in \mathbb{R}^{n\times n}$ is indexed in ascending order, i.e., $0 \leq \lambda_1(\mathbf{X}) \leq \cdots \leq \lambda_n(\mathbf{X})$; similarly, singular values of a square matrix $\mathbf{X}$ are indexed from the smallest to the largest as $0 \leq \sigma_1(\mathbf{X}) \leq \cdots \leq \sigma_n(\mathbf{X})$; and the induced 2-norm is denoted by $\|\mathbf{X}\| = \sigma_n(\mathbf{X})$. Given two positive semi-definite matrices $\mathbf{X}$ and $\mathbf{Y}$, we say that $\mathbf{X} \preceq \mathbf{Y}$ if $\mathbf{Y} - \mathbf{X}$ is positive semi-definite, and that $\mathbf{X} \prec \mathbf{Y}$ if $\mathbf{Y} - \mathbf{X}$ is positive definite. A normal random variable with mean μ and covariance matrix $\mathbf{\Sigma}$ is denoted by $\mathcal{N}(\mu, \mathbf{\Sigma})$. The expected value of a random variable is shown by $\mathbb{E}\{.\}$ and the probability of an event is denoted by $\mathbb{P}\{.\}$. The cumulative distribution function (cdf) of a scalar normal variable $\mathcal{N}(\mu, \sigma)$ is denoted by $F(x; \mu, \sigma)$. For sequences $\{a_n\}_{n\geq 1}$ and $\{b_n\}_{n\geq 1}$ with positive elements, notation $a_n = O(b_n)$ implies that a_n/b_n is bounded.

5.3 Problem Statement

We consider linear dynamical networks that consist of multiple subsystems with state vector

$$x := [x_1, \ldots, x_n]^T, \tag{5.1}$$

where $x_i \in \mathbb{R}$ is the state variable of subsystem $i \in \{1, 2, \ldots, n\}$. These subsystems are interconnected and their collective dynamics is governed by

$$\dot{x} = \mathbf{A}x \tag{5.2}$$

in which $\mathbf{A}$ is time-invariant. It is assumed that initial state $x_0 \in \mathbb{R}^n$ of the network is unknown. In order to recover the initial state, suppose that samples can only be collected from a subset of subsystems $\Omega = \{i_1, i_2, \ldots, i_p\} \subset \{1, \ldots, n\}$, where Ω is called the set of sampling locations. At every spatial location $i \in \Omega$, sensors are allowed to take finite number of samples with different time stamps; the set of such sampling times is denoted by Θ_i. A sampling strategy for subsystem $i \in \Omega$ is given by the set of ordered pairs

$$\mathfrak{S}_i = \big\{(i, t) \;\big|\; t \in \Theta_i\big\}.$$

A sampling strategy for the entire network can be obtained by

$$\mathfrak{S} = \bigcup_{i \in \Omega} \mathfrak{S}_i.$$

For a given sampling strategy $\mathfrak{S}$, the corresponding vector of samples or observations is shown by

$$y = \big[\, x_i(t) + \xi_i(t) \,\big]_{(i,t) \in \mathfrak{S}}, \tag{5.3}$$

where measurement noises $\xi_i(t)$ in all samples are assumed to be independent from each other and have normal (Gaussian) distributions with zero mean and σ^2 variance. For a given set of sampling locations Ω, the corresponding output matrix is defined by

$$\mathbf{C}_\Omega = \left[\; e_{i_1} \mid \quad \ldots \quad \mid e_{i_p} \;\right]^T. \tag{5.4}$$

Assumption 5.3.1. *The set of sampling locations Ω is chosen such that the pair $(\mathbf{A}, \mathbf{C}_\Omega)$ is observable.*

Verifying observability of a network with respect to a given set of sampling locations is an interesting and active field of research on its own, and it is different from what we investigate here. For instance, in [24], the authors adapt a graphical approach to identify those sensors that are necessary for reconstruction of the initial state. Such results may offer option for $\mathbf{A}$ and Ω that satisfy Assumption 1.

The *research problem* of this chapter is to characterize properties of sampling strategies that allow us to recover initial state of linear network (5.2) using sparse sets of samples in space and time.

5.4 Characterization of Sampling Strategies

We apply tools from finite frame theory to reformulate the observability problem and characterize its feasible sampling strategies.

5.4.1 Reconstruction in Frame Theory

The contents of this subsection are based on adjusted materials from reference [138].

Definition 5.4.1. *For a given family of vectors $(\phi_i)_{i=1,\dots,m}$ in $\mathbb{R}^n$, the corresponding analysis operator $\mathbf{T}: \mathbb{R}^n \to \mathbb{R}^m$ is defined by*

$$\mathbf{T}(x) := [\langle x, \phi_i \rangle]_{i=1,\dots,m} \tag{5.5}$$

and its frame operator $\mathbf{S}: \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$\mathbf{S}(x) := \sum_{i=1}^{m} \langle x, \phi_i \rangle \, \phi_i. \tag{5.6}$$

It is straightforward to verify that operator $\mathbf{T}$ admits the following canonical matrix representation

$$\mathbf{T} = [\phi_1 | \dots | \phi_m]^T \in \mathbb{R}^{m \times n}. \tag{5.7}$$

Thus, the canonical matrix representation of the frame operator is

$$\mathbf{S} = \mathbf{T}^T\mathbf{T} \ \in \mathbb{R}^{n\times n}. \tag{5.8}$$

Definition 5.4.2. *A family of vectors $(\phi_i)_{i=1,\dots,m}$ in $\mathbb{R}^n$ is a frame for $\mathbb{R}^n$ if there exists constants $0 < \alpha \leq \beta$ such that*

$$\alpha\|x\|_2^2 \ \leq \ \sum_{i=1}^{m} |\langle x, \phi_i\rangle|^2 \ \leq \ \beta\|x\|_2^2 \quad \textit{for all } x \in \mathbb{R}^n.$$

The largest lower frame bound and smallest upper frame bound are called the optimal frame bounds.

Proposition 5.4.3. *Let us consider a family of vectors $\Phi = (\phi_i)_{i=1,\dots,m}$ in $\mathbb{R}^n$. The following statement are equivalent:*

(i) The family of vectors Φ forms a frame for $\mathbb{R}^n$.

(ii) The set of vectors Φ span $\mathbb{R}^n$. Thus, $m = |\Phi| \geq n$.

(iii) The corresponding frame operator is positive definite, i.e., $\mathbf{S} \succ 0$, with optimal frame bounds $\alpha = \lambda_1(\mathbf{S})$ and $\beta = \lambda_n(\mathbf{S})$.

(iv) The corresponding analysis operator $\mathbf{T}$ is injective[1] with a pseudo-inverse

$$\mathbf{T}^\dagger := \left(\mathbf{T}^T\mathbf{T}\right)^{-1}\mathbf{T}^T = \mathbf{S}^{-1}\mathbf{T}^T, \tag{5.9}$$

which is a left-inverse of $\mathbf{T}$ that satisfies $\mathbf{T}^\dagger\mathbf{T} = \mathbf{I}$.

One of the well-studied problems in frame theory is to reconstruct an unknown vector $x \in \mathbb{R}^n$ from the following vector of observations

$$y = \mathbf{T}x = [\langle x, \phi_i\rangle]_{i=1,\dots,m} \in \mathbb{R}^m. \tag{5.10}$$

The following known result highlights role of $\mathbf{T}^\dagger$ in the reconstruction process from these observations.

[1] i.e., its matrix representation has full column rank.

Proposition 5.4.4. *If the family of vectors Φ forms a frame for $\mathbb{R}^n$, then any vector $x \in \mathbb{R}^n$, with a corresponding vector of observations $y \in \mathbb{R}^m$ as in (5.10), can be reconstructed via*

$$x = \mathbf{T}^{\dagger} y, \tag{5.11}$$

where $\mathbf{T}$ is the analysis operator of Φ and $\mathbf{T}^{\dagger}$ is given by (5.9).

5.4.2 Initial State Reconstruction

The solution of the linear network (5.2) is given by

$$x(t) = \mathrm{e}^{\mathbf{A}t} x_0,$$

where its i'th component is

$$x_i(t) = e_i^T \mathrm{e}^{\mathbf{A}t} x_0 = \left\langle x_0 \ , \mathrm{e}^{\mathbf{A}^T t} e_i \right\rangle. \tag{5.12}$$

Based on the definition of a frame, (5.12) reveals that the following families of vectors are the only candidates for building constructors to recover initial state of the network.

Theorem 5.4.5. *Suppose that Ω is the set of sampling locations and Θ_i is the set of sampling times for each location $i \in \Omega$. Every initial state of linear network* (5.2) *can be reconstructed from the set of samples that are collected according to sampling strategy $\mathfrak{S} = \{(i,t) |\ i \in \Omega,\ t \in \Theta_i\}$ if and only if the family of vectors*

$$\Phi(\mathbf{A}, \mathfrak{S}) = \left(\mathrm{e}^{\mathbf{A}^T t} e_i \ \middle|\ (i,t) \in \mathfrak{S} \right) \tag{5.13}$$

is a frame for $\mathbb{R}^n$.

The conclusion in Theorem 5.4.5 asserts that initial state of the network can be recovered from the vector of observations

$$y = \mathbf{T} x_0 = [x_i(t)]_{(i,t) \in \mathfrak{S}}$$

using the following equation

$$x_0 = \mathbf{T}^{\dagger} y,$$

where $\mathbf{T}$ is the analysis matrix of frame (5.13).

Remark 15. A frame for $\mathbb{R}^n$ must contain at least n vectors. Hence, the number of components in frame (5.13) satisfies

$$\sum_{i \in \Omega} |\Theta_i| \geq n.$$

This inequality implies that taking less spatial samples should be compensated by taking more temporal samples. This hints at an inherent tradeoff between the minimum number of samples in space and time required for a successful initial state reconstruction.

It turns out that observability at the sampling locations is a necessary condition for the reconstruction problem.

Lemma 5.4.6. *Suppose that the family of vectors* (5.13) *forms a frame for* $\mathbb{R}^n$*. Then, the pair* $(\mathbf{A}, \mathbf{C}_\Omega)$ *is observable.*

This can be interpreted as follows: if the sampling locations Ω create an unobservable output matrix $\mathbf{C}_\Omega$, then the initial state reconstruction will be always infeasible independent of the number of time samples.

In the rest of the paper, whenever it is not ambiguous, we drop argument of $\Phi(\mathbf{A}, \mathfrak{S})$ in (5.13) and simply write Φ. Whenever (5.13) forms a frame, it will be referred to as an observability frame. The space of all observability frames in $\mathbb{R}^n$ is denoted by $\mathfrak{F}$.

5.5 Estimation Measures

In the previous section, the reconstruction problem was formulated in noise absence. One needs to solve an estimation problem when measurement noise is presented, which requires some appropriate mechanism to measure quality of the resulting estimations. We start this section by showing that some useful estimation measures can be quantified in terms of the frame eigenvalues (i.e., eigenvalues of the frame matrix) .

5.5.1 Estimation Measures

Instead of pure measurements (6.12), suppose that a noisy observation vector is collected

$$\hat{y} = y + \xi \in \mathbb{R}^m, \tag{5.14}$$

in which $\xi \in \mathbb{R}^m$ is a zero mean Gaussian measurement noise with independent components and covariance $\mathbb{E}\left\{\xi\xi^T\right\} = \sigma^2\mathbf{I}$. For the linear network (5.2), the equation (5.14) can be rewritten in the following form,

$$\hat{y} = \mathbf{T}x_0 + \xi, \tag{5.15}$$

where $\mathbf{T}$ is the analysis matrix associated with the observability frame Φ in (5.13). Let us denote an estimation of x_0 by $\hat{x}_0$ and define the corresponding estimation error as

$$\eta := \hat{x}_0 - x_0. \tag{5.16}$$

Definition 5.5.1. *An operator $\rho : \mathfrak{F} \to \mathbb{R}$ is called (decreasingly) monotone if $\rho(\Phi_2) \leq \rho(\Phi_1)$ for all $\Phi_1 \subseteq \Phi_2$.*

In the following, we discuss two common estimation measures to compare different observability frames.

(i) Standard Deviation of the Estimation Error: For a given noisy observation vector (5.15) with underlying observability frame Φ, this estimation measure is defined by

$$\rho_d(\Phi) := \sqrt{\mathbb{E}\{\|\eta\|_2^2\}}.$$

This measure has been widely used to compute an optimal estimation via least-squares approximation [139].

Proposition 5.5.2. *Suppose that a noisy observation vector $\hat{y}$ as in (5.15) is given. Then,*

$$\hat{x}_0 = \mathbf{T}^\dagger\hat{y} \tag{5.17}$$

is an unbiased estimator for x_0 *with* $\mathbb{E}\{\hat{x}_0\} = x_0$ *that minimizes* $\|\mathbf{T}\hat{x}_0 - \hat{y}\|_2$. *Moreover, the (least-squares) estimation measure* $\rho_d : \mathfrak{F} \to \mathbb{R}_+$ *is monotone and can be characterized as*

$$\rho_d(\Phi) = \sigma \left(\sum_{i=1}^{n} \lambda_i(\mathbf{S})^{-1} \right)^{1/2} \tag{5.18}$$

where $\lambda_1(\mathbf{S}), \ldots, \lambda_n(\mathbf{S})$ *are eigenvalues of the corresponding frame operator* $\mathbf{S}$.

(ii) Differential Entropy of the Estimation Error: Since the measurement noise in (5.15) is assumed to be an independent Gaussian random variable $\mathcal{N}(0, \sigma^2\mathbf{I})$, one can use (5.17) to show that the estimation error η is also a normal random variable

$$\eta \sim \mathcal{N}(0, \sigma^2\mathbf{S}^{-1}). \tag{5.19}$$

The differential entropy of random variable $\eta \in \mathbb{R}^n$ with probability density function $p(\eta)$ is defined as

$$h(\eta) := \int_{\mathbb{R}^n} p(\eta) \log p(\eta) \; d\eta.$$

For Gaussian measurement noises, h quantifies the uncertainty volume of the estimation error.

Proposition 5.5.3. *Under the Gaussian measurement noise assumption, the value of differential entropy of the estimation error is given by*

$$h(\eta) = \frac{1}{2}\rho_e(\Phi) + \frac{n}{2}\big(1 + \log(2\pi\sigma^2)\big)$$

with

$$\rho_e(\Phi) = -\sum_{i=1}^{n} \log\big(\lambda_i(\mathbf{S})\big). \tag{5.20}$$

Moreover, the above operator $\rho_e : \mathfrak{F} \to \mathbb{R}$ *is monotone.*

5.5.2 Effects of Dwell-Time on Quality of Estimation

Suppose that sensors are scheduled to take samples according to a sampling strategy $\mathfrak{S}$, but actual measurements are taken with a uniform dwell time $\delta \in \mathbb{R}$. Let us represent the

resulting family of vectors by

$$\Phi_\delta = \Big(\mathrm{e}^{\mathbf{A}^T(t+\delta)} e_i \;\Big|\; (i,t) \in \mathfrak{S} \Big). \tag{5.21}$$

One can equivalently represent this set using (5.13) as

$$\Phi_\delta = \Big(\mathbf{B}_\delta \, \phi \;\Big|\; \phi \in \Phi \Big),$$

where $\mathbf{B}_\delta := \mathrm{e}^{\mathbf{A}^T \delta}$ is full rank for all $\delta \in \mathbb{R}$. It is straightforward to verify that elements of Φ_δ span $\mathbb{R}^n$ if and only if the elements of Φ span $\mathbb{R}^n$. Thus, the family of vectors Φ_δ is a frame for $\mathbb{R}^n$ if and only if Φ forms a frame for $\mathbb{R}^n$. The next result shows that the estimation quality is not shift-invariant.

Proposition 5.5.4. *Suppose that measurement noise (5.14) has normal distribution* $\mathcal{N}(0, \sigma^2 \mathbf{I})$. *Then,*

$$\rho_d(\Phi_\delta) \;\leq\; \sigma \Big(\sum_{i=1}^{n} \sigma_i^2 \big(\mathrm{e}^{-\mathbf{A}\delta} \big) \, \lambda_i(\mathbf{S})^{-1} \Big)^{1/2} \tag{5.22}$$

and

$$\rho_e(\Phi_\delta) = \rho_e(\Phi) - \sum_{i=1}^{n} \log \Big(\sigma_i^2 \big(\mathrm{e}^{-\mathbf{A}\delta} \big) \Big) \tag{5.23}$$

in which σ_i*'s are the singular values of the corresponding matrix.*

The upper bound (5.22) becomes tight for $\delta = 0$ because $\sigma_i(\mathrm{e}^{\mathbf{A}\delta}) = \sigma_i(\mathbf{I}) = 1$ for all $i = 1, \ldots, n$. When $\mathbf{A}$ is Hurwitz, according to inequality (5.22), the estimation quality deteriorates as $\delta > 0$ gets larger. The reason is that magnitude of samples decrease and the measurement noise (with constant intensity) becomes more dominant as time goes by. In fact, (5.22) implies that anti-stable state matrices neutralize negative effects of dwell time on the quality of estimation.

5.6 Construction of Observability Frames

Let us represent distinct eigenvalues of state matrix $\mathbf{A}$ by distinct eigenvalues $\lambda_1(\mathbf{A}), \ldots, \lambda_q(\mathbf{A})$ for some $q \leq n$ and its corresponding minimal polynomial[2] by

$$\mathrm{p}_{\mathbf{A}}(\lambda) = \prod_{m=1}^{q} \big(\lambda - \lambda_m(\mathbf{A})\big)^{p_m} \tag{5.24}$$

for some positive integers p_m, whose degree is denoted by $\mathfrak{d}(\mathbf{A})$ which is less than or equal to n. To state our next result, we need to define the row vector map

$$E(t) := \Big[\mathrm{e}^{\lambda_m(\mathbf{A})t} t^k\Big]_{\substack{m=1,\ldots,q \\ k=0,\ldots,p_m-1}} \in \mathbb{R}^{1\times\mathfrak{d}(\mathbf{A})}. \tag{5.25}$$

Theorem 5.6.1. *Suppose that a sampling strategy $\mathfrak{S} = \{(i,t)|\ i \in \Omega,\ t \in \Theta_i\}$ is adopted such that:*

- *At every $i \in \Omega$, $M_i := |\Theta_i| \geq \mathfrak{d}(\mathbf{A})$ samples are collected,*
- *$\mathbf{E}_i$ has full column rank, where*

$$\mathbf{E}_i := [E(t)]_{t\in\Theta_i} \in \mathbb{R}^{M_i\times\mathfrak{d}(\mathbf{A})}. \tag{5.26}$$

Then, under Assumption 5.3.1, the family of vectors

$$\Phi = \Big(\mathrm{e}^{\mathbf{A}^T t} e_i \ \Big|\ (i,t) \in \mathfrak{S}\Big) \tag{5.27}$$

forms a frame for $\mathbb{R}^n$.

Any frame for $\mathbb{R}^n$ must have at least n vectors. Hence, prior to the application of Theorem 5.6.1, a necessary condition for the total number of sampling times is

$$|\mathfrak{S}| = \sum_{i\in\Omega} |\Theta_i| \geq n. \tag{5.28}$$

On the other hand, Theorem 5.6.1 requires $|\Theta_i| \geq \mathfrak{d}(\mathbf{A})$. Comparing these two arguments

[2]The minimal polynomial of matrix $\mathbf{A}$ is the monic polynomial in $\mathbf{A}$ of smallest degree such that $\mathrm{p}_{\mathbf{A}}(\mathbf{A}) = 0$. This should not be confused with the characteristic polynomial of a matrix, which is always of degree n and only in certain cases coincides with the minimal polynomial [140].

implies that the resulting frame Φ from Theorem 5.6.1 will have many redundant elements as the number of locations $|\Omega|$ increases. This motivates our investigation in the next section to seek scalable algorithms to construct sparse frames (in space and time) out of highly redundant observability frames.

According to Theorem 5.6.1, the sufficient number of samples at each location is n. This condition is rather conservative as it takes into accounts situations where samples are taken only from a very small (compared to n) subset of spatial locations. In Theorem 5.6.7, it is shown that if $|\Omega| = n$, then we may collect as few as one sample from each spatial location.

The set of all time instances for which $\mathbf{E}_i$ is not full column rank have zero Lebesgue measure in the corresponding design space. In fact, Theorem 5.6.1 suggests that one can comfortably skip rank verification step.

Corollary 5.6.2. *For a given $\tau > 0$, suppose that the sampling times in Theorem 5.6.1 are drawn randomly and independently from the uniform distribution over $[0, \tau]$. Then, with probability 1, the family of vectors in (5.27) is a frame for $\mathbb{R}^n$.*

Remark 16. Theorem 5.6.1 does not directly advise us to choose certain locations and times for an optimal estimation quality. Nevertheless, the latter corollary motivates an approach for finding a sparse sampling strategy with an acceptable quality. First, we can randomly construct a rich and *dense* set of space-time sampling indices. Then, we can use sparsification to discover the pivotal components of the sampling strategy (see next section).

Example 5.6.3. Let us consider the two-dimensional system

$$\dot{x} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} x. \tag{5.29}$$

We choose to sample only from the first subsystem; i.e., $\Omega = \{1\}$, $\mathbf{C}_\Omega = \begin{bmatrix} 1 & 0 \end{bmatrix}$. Thus, $(\mathbf{A}, \mathbf{C}_\Omega)$ is observable and

$$\mathrm{e}^{\mathbf{A}t} = \begin{bmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{bmatrix}.$$

Let us pick sampling times $t_1, t_2 \in [0, \tau]$, i.e., $M_1 = 2$. The corresponding family of vectors is

$$\Phi = \left(\mathrm{e}^{\mathbf{A}^T t_1} e_1, \mathrm{e}^{\mathbf{A}^T t_2} e_1\right) = \left(\begin{bmatrix} \cos(t_1) \\ -\sin(t_1) \end{bmatrix}, \begin{bmatrix} \cos(t_2) \\ -\sin(t_2) \end{bmatrix} \right).$$

In this case, matrix (5.26) is

$$\mathbf{E}_1 = \begin{bmatrix} \mathrm{e}^{\mathrm{j}t_1} & \mathrm{e}^{-\mathrm{j}t_1} \\ \mathrm{e}^{\mathrm{j}t_2} & \mathrm{e}^{-\mathrm{j}t_2} \end{bmatrix} \;\Rightarrow\; \det(\mathbf{E}_1) = 2\mathrm{j}\sin(t_1 - t_2).$$

Hence, according to Theorem 5.6.1, if $t_1 - t_2 \neq k\pi$ for $k \in \mathbb{Z}$, then Φ is a frame. Alternatively, if we compute the frame matrix $\mathbf{S}$, using trigonometric identities, we get

$$\det(\mathbf{S}) = \sin^2(t_1 - t_2),$$

which gives us the same constraints on the sampling times. Since the Lebegues measure of the points for which $\sin^2(t_1 - t_2) = 0$ is indeed zero in $[0, \tau] \times [0, \tau]$, any random choices for t_1 and t_2 will result into a frame with probability 1. The latter observation agrees with Corollary 5.6.2. Next, we consider sampling $M_1 = M$ samples at location 1 for $M > 2$. By induction on sampling times $\Theta_1 = \{t_1, \ldots, t_M\}$, we have

$$\det(\mathbf{S}) = \sum_{\substack{i=1,\ldots,M \\ j=i+1,\ldots,M}} \sin^2(t_i - t_j). \tag{5.30}$$

Again, if $t_i - t_j \neq k\pi$ for $k \in \mathbb{Z}$, we get a frame out of these observations. Random sampling also results in a frame with probability 1. Moreover, $\rho_d(\Phi) = \sqrt{2}\sigma/\sqrt{\det(\mathbf{S})}$ and it is shift-invariant.

Now, we briefly look at periodic sampling strategy,[3] i.e., $\Theta_i = \{0, \delta, \ldots, (M_i - 1)\delta\}$ for every $i \in \Omega$, where $\delta > 0$ is a known sampling step-size.

[3] practical implication of this strategy is that sensors take samples with some certain frequency based on a synchronized digital clock.

Theorem 5.6.4. *Suppose that the sampling step-size satisfies*

$$\big(\lambda_m(\mathbf{A}) - \lambda_{m'}(\mathbf{A})\big)\,\delta \notin 2\pi\mathrm{j}\mathbb{Z} \tag{5.31}$$

for all distinct eigenvalues $\lambda_m(\mathbf{A})$ and $\lambda_{m'}(\mathbf{A})$ of the state matrix $\mathbf{A}$. If $M_i \geq \mathfrak{d}(\mathbf{A})$, then the family of vectors

$$\Phi = \Big(\mathbf{B}_\delta^k e_i \;\Big|\; i \in \Omega,\ k = 0, \ldots, M_i - 1\Big) \tag{5.32}$$

forms a frame for $\mathbb{R}^n$, where $\mathbf{B}_\delta := \mathrm{e}^{\mathbf{A}^T\delta}$.

Example 5.6.5 (Example 5.6.3 continued). A sufficient condition for the sampling step-size for linear system (5.29) is

$$\mathrm{j} - (-\mathrm{j})\delta \;=\; 2\mathrm{j}\delta \;\notin\; 2\pi\mathrm{j}\;\mathbb{Z} \;\Rightarrow\; \delta \notin \pi\mathbb{Z}.$$

Alternatively, because $t_i - t_j = (i-j)\delta$, using the expression for $\det(\mathbf{S})$ in (5.30), Φ is a frame if $\delta \notin \pi\mathbb{Z}$.

For a state matrix $\mathbf{A}$ whose all eigenvalues are real, one may verify that the requirement (5.32) for any positive step-size δ. Therefore we have the following corollary by Theorem 5.6.4.

Corollary 5.6.6. *If all eigenvalues of $\mathbf{A}$ are real, then for every step-size $\delta > 0$ and sampling horizon $M_i \geq \mathfrak{d}(\mathbf{A}), i \in \Omega$, the family of vectors (5.32) is a frame for $\mathbb{R}^n$.*

Now, we consider the case where collecting samples from all locations is possible and samples are taken in a small time range.

Theorem 5.6.7. *Suppose that $\Omega = \{1, \ldots, n\}$ is the set of sampling locations and the set of sampling times Θ_i, for each sampling location $i \in \Omega$, is chosen such that $|\Theta_i| = M_i \geq 1$ and*

$$[t^*, t^* + \delta^*) \cap \Theta_i \neq \varnothing \tag{5.33}$$

for some $t^* \in \mathbb{R}$, *where step-size* $\delta^* > 0$ *is given by*

$$\delta^* < (\ln 2)\,\|\mathbf{A}\|^{-1}. \tag{5.34}$$

Then, the sampling strategy

$$\mathfrak{S} = \Big\{(i,t) \;\Big|\; i = 1,\ldots,n \text{ and } t \in \Theta_i\Big\}$$

results in a family of vectors

$$\Phi = \Big(\mathrm{e}^{\mathbf{A}^T t} e_i \;\Big|\; (i,t) \in \mathfrak{S}\Big) \tag{5.35}$$

that forms a frame for $\mathbb{R}^n$.

The time range δ^* in Theorem 5.6.7 only depends on the state matrix $\mathbf{A}$ and is strictly positive. Next, we consider the case where collecting samples from all locations is possible and samples are taken randomly in a time range $[0,\tau]$, which is not necessarily in a small time range.

Corollary 5.6.8. *Suppose that samples are collected from all subsystems, i.e.,* $\Omega = \{1,\ldots,n\}$, *at least once, i.e.,* $|\Theta_i| \geq 1$. *Sampling times are drawn randomly and independently from the uniform distribution over interval* $[0,\tau]$. *Then, the resulting family of vectors* (5.35) *is a frame for* $\mathbb{R}^n$ *with probability* 1.

Example 5.6.9 (Example 5.6.3 continued). For the linear system (5.29), let us consider a full state sampling with strategy $\mathfrak{S} = \big\{(1,t_1),(2,t_2)\big\}$ that results in vectors

$$\Phi = \Big(\mathrm{e}^{\mathbf{A}^T t_1} e_1, \mathrm{e}^{\mathbf{A}^T t_2} e_2\Big) = \left(\begin{bmatrix}\cos(t_1)\\ -\sin(t_1)\end{bmatrix}, \begin{bmatrix}\cos(t_2)\\ \sin(t_2)\end{bmatrix}\right). \tag{5.36}$$

For the corresponding frame matrix, we have

$$\det(\mathbf{S}) = 1 - \sin^2(t_1 - t_2). \tag{5.37}$$

Thus, Φ is a frame for $\mathbb{R}^2$ if and only if

$$t_1 - t_2 \neq \left(k + \frac{1}{2}\right)\pi \quad \text{for all} \quad k \in \mathbb{Z}. \tag{5.38}$$

Alternatively, δ^* in Theorem 5.6.7 satisfies

$$\delta^* < \ln 2.$$

According to Theorem 5.6.7, if sampling times t_1 and t_2 satisfy

$$|t_1 - t_2| < \ln 2, \tag{5.39}$$

then the family of vectors (5.36) is a frame for $\mathbb{R}^n$. Comparing the two constraints characterized by (5.39) and (5.38) reveals that the resulting condition for sampling times from Theorem 5.6.7 is more conservative. To verify effectiveness of Corollary 5.6.8, one can verify that the Lebesgue measure of all pairs of points $\{t_1, t_2\}$ in $[0, \tau]^2 \subset \mathbb{R}^2$ for which $\det(\mathbf{S}) = 0$ is zero. As a result, one can randomly select sampling times $\{t_1, t_2\}$ to construct a frame with probability 1.

5.7 Frame Sparsification

Suppose that a highly redundant set of samples from network (5.2) is provided for the estimation problem. This usually happens when a conservative sampling strategy $\mathfrak{S}$ is used and all subsystems are allowed to collect numerous samples over time. Even if the corresponding family of vectors

$$\Phi = \left(\mathrm{e}^{\mathbf{A}^T t} e_i \;\middle|\; (i,t) \in \mathfrak{S}\right) \tag{5.40}$$

forms a frame, i.e., the resulting estimation problem is feasible, there are still unnecessary and undesired degrees of redundancy that should be trimmed away in order to enhance scalability properties of the estimation algorithms.

Definition 5.7.1. *For a given design parameter $\theta > 0$, a family of vectors Φ_s in $\mathbb{R}^n$ is called a θ-approximation of frame Φ if:*

(i) $\Phi_s \subset \Phi$ and it has at most $\theta|\Omega|$ elements,

(ii) Φ_s is a frame for $\mathbb{R}^n$.

For a given (highly redundant) frame (5.40) and some parameter $\theta > 0$, our goal is to find a sampling strategy $\mathfrak{S}_s$ whose corresponding family of vectors

$$\Phi_s = \left(\mathrm{e}^{\mathbf{A}^T t} e_i \;\middle|\; (i,t) \in \mathfrak{S}_s \right)$$

is a $\theta-$approximation of (5.40). Condition (i) mandates the sampling strategy $\mathfrak{S}_s$ to collect at most θ samples in average from all subsystems, i.e.,

$$\bar{\Theta}(\mathfrak{S}_s) = \frac{1}{|\Omega|} \sum_{i \in \Omega} |\Theta_i(\mathfrak{S}_s)| = \frac{|\mathfrak{S}_s|}{|\Omega|} \leq \frac{|\Omega|\theta}{|\Omega|} = \theta.$$

Condition (ii) ensures the feasibility of the resulting estimation problem. In the following, we will discuss three methods to achieve our goal.

Remark 17. Sparsity is a relative notion. In this chapter, it is reasonable to consider a sampling strategy to be sparse if it takes almost linear number of samples (in terms of network size) in a given time window. This is the level of sparsity that the main result of this section, Theorem 7.5.2, achieves; we refer to next subsection.

5.7.1 Sparsification by Leverage Scores

Our first approach is based on sparsification via the notion of effective resistances [51], which depends on the concentration properties of the sums of random outer-products and was originally developed for sparsification of weighted graph Laplacians. Similar to graph Laplacian, the frame matrix $\mathbf{S}$ is also a sum of rank-one matrices

$$\mathbf{S} = \sum_{\phi \in \Phi} \phi \phi^T.$$

Definition 5.7.2. *For a given finite frame Φ, the leverage scores are positive numbers that*

are defined by

$$r_\phi(\mathbf{S}) := \phi^T \mathbf{S}^{-1} \phi \tag{5.41}$$

for every $\phi \in \Phi$.

One can associate a probability mass function $\pi : \Phi \to [0, 1]$ to a given frame Φ using its leverage scores as follows:

$$\pi(\phi) = \frac{r_\phi(\mathbf{S})}{n}. \tag{5.42}$$

This gives a well-defined mass function as

$$\sum_{\phi \in \Phi} \pi(\phi) = \frac{1}{n} \sum_{\phi \in \Phi} \mathrm{Tr}\Big(\mathbf{S}^{-1} \phi \phi^T\Big) = 1.$$

As it is summarized in Algorithm 4, elements of Φ are sampled iteratively and independently, with replacement, according to probability mass function (5.42). A sampled element will be added to Φ_s if it is not already in Φ_s. The resulting sparsified frame Φ_s will have at most q elements. One may estimate $|\Omega_s|$, i.e., the number of sampling locations after sparsification, and obtain a reasonable estimate for $\theta \leq q/|\Omega_s|$. Algorithm 4 also assigns a weight to every elements of Φ_s via weight function $w_s : \Phi \to \mathbb{R}_+$. Each execution of Algorithm 4 returns a different realization of w_s, where $w_s(\phi) = 0$ for $\phi \notin \Phi_s$. These weights are useful in quantifying estimation-quality loss due to sparsification. In fact, the weight function w_s is a bounded random variable, where $w_s(\phi)$ may assume different realizations drawn from $\big\{p(q\pi(\phi))^{-1} \;\big|\; p = 0, 1, \ldots, q\big\}$.

Theorem 5.7.3. *For a given frame* Φ *in* $\mathbb{R}^n$*, let us fix parameter* $\epsilon \in (1/\sqrt{n}, 1]$ *and the number of samples* $q = O(n \log n/\epsilon^2)$*. Then, the resulting set of elements* Φ_s *from Algorithm 4 is also a frame for* $\mathbb{R}^n$ *with probability at least* $1/2$*. Furthermore, with probability at least* $1/4$*, the estimation-quality losses satisfy*[4]

$$\frac{\rho_d(\Phi_s) - \rho_d(\Phi)}{\rho_d(\Phi)} \leq -1 + \sqrt{\frac{4\,\bar{\chi}}{1-\epsilon}} \tag{5.43}$$

$$\rho_e(\Phi_s) - \rho_e(\Phi) \leq n \log\left(\frac{4\,\bar{\chi}}{1-\epsilon}\right), \tag{5.44}$$

[4]Using monotonicity property of the estimation measures, one can show that upper bounds in (7.32) and (7.33) are nonnegative.

Algorithm 4 Randomized Frame Sparsification

input: frame $\Phi = (\phi_1, \ldots, \phi_{|\mathfrak{S}|})$ and design parameters $q, \epsilon > 0$
output: set of vectors Φ_s and weight function w_s

initialize: $\Phi_s = \varnothing$, $w_s(.) = 0$, $\mathbf{S}_s = \mathbf{0}$
  for $k = 1$ to q **do**
    sample an element from Φ with probability distribution $\pi \to \phi$
    update weight function : $w_s(\phi) \leftarrow w_s(\phi) + (q\pi(\phi))^{-1}$
    if $\phi \notin \Phi_s$, **then**
      add ϕ to Φ_s
      update the frame matrix: $\mathbf{S}_s \leftarrow \mathbf{S}_s + \phi\phi^T$
    end if
  end for

where $\bar{\chi} := \mathbb{E}\{\chi\}$ *and* χ *is a random variable given by*

$$\chi := \inf\left\{\gamma > 0 \;\middle|\; \sum_{\phi\in\Phi} w_s(\phi)\big(\gamma - w_s(\phi)\big)\,\phi\phi^T \succeq \mathbf{0}\right\}. \tag{5.45}$$

The backbone of this result is based on Theorem 1 of [51] and asserts that Algorithm 6 trims off a given (highly redundant) frame and returns, with probability more than 0.5, a new frame whose size is almost linear in network size. Moreover, it is shown that, with probability at least 0.25, the estimation measures of the new (sparsified) frame stays within constant multiples/difference of the estimation measure of the original (redundant) frame.

Corollary 5.7.4. *The random variable* χ*, which is defined by* (5.45)*, satisfies*

$$\chi \leq \max_{\phi\in\Phi} w_s(\phi)$$

almost surely. Moreover, under the settings of Theorem 5.7.3, inequalities

$$\frac{\rho_d(\Phi_s) - \rho_d(\Phi)}{\rho_d(\Phi)} \;\leq\; -1 + \sqrt{\frac{4\,w_{\max}}{1-\epsilon}} \tag{5.46}$$

$$\rho_e(\Phi_s) - \rho_e(\Phi) \;\leq\; n\log\left(\frac{4\,w_{\max}}{1-\epsilon}\right) \tag{5.47}$$

holds with probability at least $1/4$*, where*

$$w_{max} := \mathbb{E}\left\{\max_{\phi\in\Phi} w_s(\phi)\right\}.$$

This corollary shows that there exists a clear relationship (even if it is not tight) between the performance loss and the magnitude of parameter $w_{\max}$.

The leverage scores disclose the importance of every component with respect to the entire frame for the sake of estimation. For instance, if the network is asymptotically stable, a component with a relatively large time label is expected to have a relatively small leverage score. Thus, such insignificant components are less likely to be sampled by Algorithm 4 and can be trimmed off to achieve a comparable estimation quality.

Running Time Analysis: Computing the inverse of $\mathbf{S}$ can be done in $O(n^3)$ operations, while computing the leverage scores using this matrix requires $O(|\Phi|n^2)$. We need to check for repeated samples, which does not increase the running time of the algorithm. Computing the frame matrix $\mathbf{S}_s$ can be done in $O(n \log n/\epsilon^2 \times n^2) = O(n^3 \log n/\epsilon^2)$ operations. Hence, the total running time of Algorithm 6 is $O(n^3 \log n/\epsilon^2 + |\Phi|n^2)$.

5.7.2 Random Partitioning and Kadison-Singer Paving Solution

If the leverage scores (5.41) are uniformly bounded by a small enough number, then components of a frame can be partitioned in a balanced manner in order to obtain two separate subframes with explicit bounds on their spectra. Such spectral bounds are useful to find bounds on the estimation quality of the resulting subframes. The next theorem is based on Corollary 1.3 of the recent seminal paper [136] that gives us a paving solution to the famous Kadison-Singer problem.

Proposition 5.7.5. *Suppose that the leverage scores in* (5.41) *satisfy*

$$r_\phi(\mathbf{S}) \leq r^* \tag{5.48}$$

for all $\phi \in \Phi$ *and some positive number* $r^* < 1.5 - \sqrt{2}$. *Let us randomly partition* Φ *into two subfamilies* Φ_1 *and* Φ_2 *such that every element of* Φ, *independent of others, belongs to either of the partitions with probability* $1/2$. *Then, with a positive probability, the resulting*

partition will satisfy

$$\left(1-\frac{(1+\sqrt{2r^*})^2}{2}\right)\mathbf{S} \preceq \sum_{\phi\in\Phi_j}\phi\phi^T \preceq \left(1+\frac{(1+\sqrt{2r^*})^2}{2}\right)\mathbf{S}$$

for $j=1$ *and* 2.

For a given (highly redundant) observability frame, the result of Proposition 5.7.5 allows us to calculate the relative/absolute estimation-quality degradation of the resulting partitions.

Theorem 5.7.6. *Suppose that the leverage scores* (5.41) *satisfy* (5.48). *Then, the randomly partitioned subfamilies* Φ_1 *and* Φ_2 *from Proposition 5.7.5 are both frames for* $\mathbb{R}^n$ *with a positive probability and the estimation-quality can be bounded as follows*

$$\frac{\rho_d(\Phi_j)-\rho_d(\Phi)}{\rho_d(\Phi)} \leq \kappa(r^*) \tag{5.49}$$

$$\rho_e(\Phi_j)-\rho_e(\Phi) \leq -n\log\left(1-\frac{(\sqrt{2r^*}+1)^2}{2}\right) \tag{5.50}$$

for $j=1,2$, *where*

$$\kappa(r) := \left(1-\frac{(\sqrt{2r}+1)^2}{2}\right)^{-1/2}-1. \tag{5.51}$$

The quantity $\kappa(r^*)$ is a worst-case bound on the relative performance degradation of the randomly partitioned subframes Φ_1 and Φ_2. This function has been illustrated in Fig. 5.1. In simulations, we observe that comparably better bounds are achievable.

Applicability of Random Partitioning: When we deal with massive incoming samples (data) from the sensors, we expect the leverage scores to be small. In fact, we observe that the leverage scores satisfy

$$\sum_{\phi\in\Phi} r_\phi(\mathbf{S}) = n \;\Rightarrow\; \bar{r}(\mathbf{S}) = \frac{n}{|\Phi|},$$

where $\bar{r}(.)$ stands for the average of the leverage scores. Hence, as a rule of thumb, one

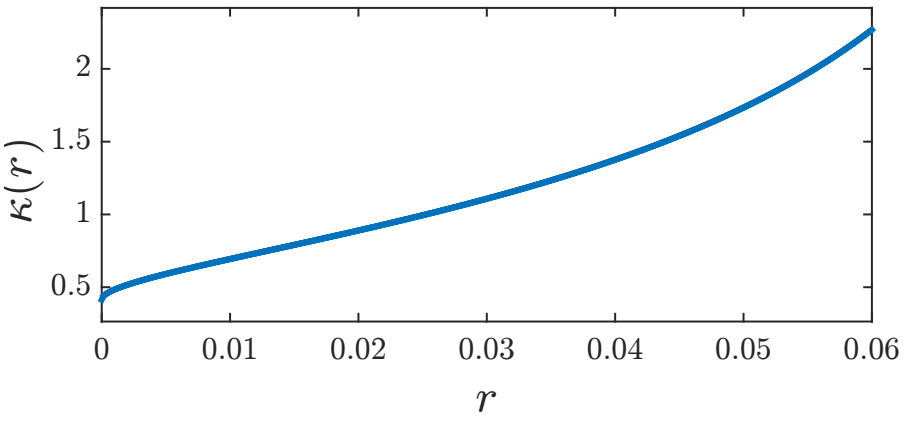

Figure 5.1: The worst case relative performance degradation based on the bound of Theorem 5.7.6.

should expect that for fairly balanced observability frames with size

$$|\Phi| > \lceil (6 + 4\sqrt{2})n \rceil,$$

the leverage score of a typical component in Φ, in average, is less than $(6 + 4\sqrt{2})^{-1} = 1.5 - \sqrt{2} \approx 0.0858$.

Running Time Analysis: Computing the frame matrix for each partition can be done in $O(|\Phi|n^2)$. Hence, the random partitioning can be done in $O(|\Phi|n^2)$ operations.

5.7.3 Greedy Sparsification

In order to maintain a predetermined level of estimation quality and sparsity, one may consider using greedy algorithms that have been demonstrated to be useful in practice with satisfactory performance for a broad range of combinatorial problems [55, 141]. In greedy frame sparsification, the core idea at every iteration is to eliminate that component of the frame which will increase value of a given estimation measure less than the others. At iteration k, let us denote the remaining frame and its frame matrix by Φ_k and $\mathbf{S}_k$, respectively. Eliminating a component ϕ from Φ_k corresponds to the following rank-one

Algorithm 5 Greedy Frame Sparsification

input: frame Φ and its frame matrix $\mathbf{S} \succ 0$, $\mathfrak{s} \in (0,1)$, $\mathfrak{e} > 0$
output: frame Φ_s
initialize: $\mathbf{S}_s = \mathbf{S}$, $\Phi_s = \Phi$

while $\dfrac{\rho_\square(\Phi_s) - \rho_\square(\Phi)}{\rho_\square(\Phi)} \leq \mathfrak{e}$ and $\dfrac{|\Phi_s|}{|\Phi|} \geq \mathfrak{s}$ **do**

find minimizer $\phi_\square^*$ via solving (5.56) or (5.57)
update the frame matrix

$$\mathbf{S}_s^{-1} \leftarrow \mathbf{S}_s^{-1} + \frac{\mathbf{S}_s^{-1}\phi_\square^*\phi_\square^{*T}\mathbf{S}_s^{-1}}{1 - \phi_\square^{*T}\mathbf{S}_s^{-1}\phi_\square^*}$$

update $\Phi_s \leftarrow \Phi_s \backslash \phi_\square^*$

end while

update

$$\mathbf{S}_{k+1} = \mathbf{S}_k - \phi\phi^T \tag{5.52}$$

with $\mathbf{S}_0 = \mathbf{S}$. According to the Sherman-Morrison formula [142], one gets update rule

$$\mathbf{S}_{k+1}^{-1} = \mathbf{S}_k^{-1} + \frac{\mathbf{S}_k^{-1}\phi\phi^T\mathbf{S}_k^{-1}}{1 - \phi^T\mathbf{S}_k^{-1}\phi}. \tag{5.53}$$

Proposition 5.7.7. *Upon eliminating a component ϕ from an observability frame Φ_k, the estimation measures are updated according to*

$$\rho_d(\Phi_{k+1}) = \sqrt{\rho_d^2(\Phi_k) + \frac{\sigma^2\, r_\phi(\mathbf{S}_k^2)}{1 - r_\phi(\mathbf{S}_k)}} \tag{5.54}$$

$$\rho_e(\Phi_{k+1}) = \rho_e(\Phi_k) - \log\big(1 - r_\phi(\mathbf{S}_k)\big). \tag{5.55}$$

At every iteration, the optimizer of $\rho_d(\Phi_{k+1})$ can be determined by solving the optimization problem

$$\phi_d^* = \arg\min_{\phi \in \Phi_k} \frac{\left\|\mathbf{S}_k^{-1}\phi\right\|^2}{1 - \phi^T\mathbf{S}_k^{-1}\phi} \tag{5.56}$$

and for $\rho_e(\Phi_{k+1})$ by solving

$$\phi_e^* = \arg\min_{\phi \in \Phi_k} \phi^T\mathbf{S}_k^{-1}\phi. \tag{5.57}$$

Algorithm 5 details all necessary steps to compute a sparsfication of a (redundant) observability frame, where we use notation $\square \in \{d, e\}$. The algorithm stops whenever either a desired sparsity level $\mathfrak{s} \in (0, 1)$ or a maximum allowable relative estimation error $\mathfrak{e} > 0$ has been achieved. This algorithm resembles the procedure of updating a performance measure of a linear consensus network when a new coupling link is added to the network [2]. The performance guarantees of the greedy methods in this context is a well-studied subject. In general, derivation of performance bounds heavily depends on the curvature conditions of the specific class of objective functions, e.g., sub-modularity, super-modularity or weak forms of these properties; please see [143] and references in there.

Running Time Analysis: Computing $\mathbf{S}^{-1}$ at the beginning requires $O(n^3)$. Then, at each iteration, one needs to compute and update the value of estimation measure for every vector in Φ, which takes $O(|\Phi|n^2)$. Thus, in order to achieve sparsity level $\mathfrak{s}$, one needs $O\big((1-\mathfrak{s})|\Phi| \times |\Phi|n^2\big) = O\big((1-\mathfrak{s})|\Phi|^2 n^2\big)$ operations. Since $|\Phi| \geq n$, Algorithm 5 can be implemented in $O\big((1-\mathfrak{s})|\Phi|^2 n^2\big)$. Compared to running time of the randomized sparsification $O(n^3 \log n/\epsilon^2 + |\Phi|n^2)$, the running time of the greedy method can be higher by an order of $|\Phi|$.

Remark 18. Our proposed algorithms in this section employ some results from [51,136]. The idea of sparsification via effective resistances are recently applied in controls community to obtain network abstraction as well as actuator scheduling in large-scale networked control systems [129] and [1]. The Kadison-Singer paving solution of [136] is also utilized in [129] as a method of randomized actuator scheduling.

5.8 Fundamental Limits and Tradeoffs

For a given linear network (5.2) whose state vector is sampled based on an arbitrary sampling strategy, we show that there are fundamental limits and tradeoffs on the best achievable values for the estimation measures and space-time sparsity.

Theorem 5.8.1. *Suppose that initial state of the n-dimensional linear network* (5.2) *is sampled under a sampling strategy* $\mathfrak{S}$ *with total number of samples* $|\mathfrak{S}|$*. Then, the best*

achievable estimation measures are bounded from below by constants that are quantified by

$$\rho_d(\Phi) \geq \frac{\sigma n}{\nu \sqrt{|\mathfrak{S}|}} \tag{5.58}$$

and

$$\rho_e(\Phi) \geq n \log \left(\frac{n}{\nu^2 |\mathfrak{S}|} \right), \tag{5.59}$$

where Φ *is the observability frame corresponding to* $\mathfrak{S}$ *and* $\nu := \nu(\mathbf{A}, \mathfrak{S})$ *is defined by*

$$\nu(\mathbf{A}, \mathfrak{S}) = \max_{t \in \Theta_1 \cup \cdots \cup \Theta_{|\Omega|}} \left\| \mathrm{e}^{\mathbf{A}t} \right\|. \tag{5.60}$$

The inequalities (5.58) and (5.59) give us some convenient rules of thumb about the scaling properties of the estimation measures. For instance, if $|\mathfrak{S}| = O(n^2)$, then it can be deduced from (5.58) that[5]

$$\rho_d(\Phi) \geq \frac{c\sigma}{\nu}$$

for some constant c. This implies that the estimation quality cannot be enhanced beyond a hard limit. Such limitations are important in network design as they are independent of the network size and sampling strategy. For more discussions on significant role of fundamental limits in control, we refer to [144, 145].

Theorem 5.8.2. *For a given linear network* (5.2)*, let us assume that there exists* $\delta > 0$ *such that all distinct eigenvalues of its state matrix satisfy*

$$\big(\lambda_i(\mathbf{A}) - \lambda_k(\mathbf{A})\big)\, \delta \notin 2\pi \mathrm{j}\, \mathbb{Z} \tag{5.61}$$

and it is sampled according to a sampling strategy with property[6] $\Theta_i \subset \delta \mathbb{Z}_+$ *for all* $i \in \Omega$*. If* $\mathbf{A}$ *is Hurwitz, then universal (i.e., independent of number of samples) fundamental limits*

[5] In the inequality (5.58), the value of σ/ν can be interpreted as noise-to-signal ratio because the value of ν relates to the norm of samples and σ is the standard deviation of measurement noise. If $\mathbf{A}$ is Hurwitz, ν is always a finite number. On the other hand, for unstable networks, the noise-to-signal ratio loses its significance as sampling process is prolonged. As a consequence, the value of lower bounds in the inequalities (5.58) and (5.59) are usually small(er) for unstable networks.

[6] There is a practical implication for this assumption: sensors usually take samples with some certain frequency based on a synchronized digital clock.

on the best achievable estimation measures emerge as follows

$$\rho_d(\Phi) \;\geq\; \sigma\sqrt{\mathrm{Tr}(\mathbf{Q}^{-1})}, \tag{5.62}$$

$$\rho_e(\Phi) \;\geq\; -\mathrm{Tr}\big(\log(\mathbf{Q})\big), \tag{5.63}$$

where $\mathbf{Q} \succ 0$ *is the observability Gramian, i.e., the unique solution of Lyapunov equation*

$$\mathrm{e}^{\mathbf{A}^T\delta}\,\mathbf{Q}\,\mathrm{e}^{\mathbf{A}\delta} \;-\; \mathbf{Q} \;+\; \mathbf{C}_\Omega^T\mathbf{C}_\Omega \;=\; \mathbf{0}. \tag{5.64}$$

Also, the lower bounds can be achieved if and only if $\mathfrak{S} = \Omega \times \delta\mathbb{Z}_+$.

In an exponentially stable linear network, as time goes by, the magnitude of state dwindles compared to measurement noise. For such systems, Theorem 5.8.2 predicts that estimation quality cannot be improved beyond some certain threshold even if the number of samples tends to infinity.

For a given sampling strategy $\mathfrak{S}$, the average number of samples per subsystem is quantified by

$$\bar{\Theta}(\mathfrak{S}) := \frac{1}{|\Omega|}\sum_{i\in\Omega}|\Theta_i(\mathfrak{S})|. \tag{5.65}$$

Theorem 5.8.3. *For given desired levels of estimation qualities* $\rho_d^*, \rho_e^* > 0$ *and parameter* $\nu^* > 0$*, let us consider all* n*-dimensional linear networks* (5.2) *whose pair of state matrices and sampling strategies belong to*

$$\mathfrak{N}_\square = \Big\{(\mathbf{A},\mathfrak{S}) \;\Big|\; \nu(\mathbf{A},\mathfrak{S}) = \nu^* \;\; \textit{and} \;\; \rho_\square\big(\Phi(\mathbf{A},\mathfrak{S})\big) = \rho_\square^*\Big\},$$

where $\square \in \{d, e\}$*. Intrinsic tradeoffs between the number of sampling locations and the average number of samples per subsystem transpire over* $\mathfrak{N}_\square$ *that are characterized by*

$$\bar{\Theta}(\mathfrak{S})\cdot|\Omega| \;\geq\; \left(\frac{\sigma n}{\nu^*\,\rho_d^*}\right)^2, \tag{5.66}$$

$$\bar{\Theta}(\mathfrak{S}) \cdot |\Omega| \;\geq\; \frac{n}{\nu^{*2}} \exp\left(-\frac{\rho_e^*}{n}\right). \tag{5.67}$$

The result of Theorem 5.8.3 asserts that intrinsic tradeoffs emerges among all linear networks with similar estimation quality and parameter (5.60): reducing number of sampling locations must be compensated by increasing the average number of samples per subsystem and vice versa.

5.9 Numerical Simulations

Let us consider a linear dynamical network (5.2) that consists of n subsystems, which are randomly and uniformly distributed over a square-shape spatial domain $[0,1] \times [0,1]$. The Euclidean (spatial) distance between the subsystems i and j is denoted by $\mathrm{dis}(i,j)$. If two subsystems lie in each others connectivity range, then there will be a coupling between the two subsystems and the corresponding entries in the state space will be nonzero numbers. More precisely, the state matrix $\mathbf{A} = [a_{ij}]$ is defined by

$$a_{ij} = \begin{cases} \zeta_{ij}\, \mathrm{e}^{-\mathfrak{a}\, \mathrm{dis}(i,j)^{\mathfrak{b}}} & \mathrm{dis}(i,j) \leq d \\ 0 & \mathrm{dis}(i,j) > d \end{cases}. \tag{5.68}$$

for some $d > 0$. The parameters $\mathfrak{a} > 0$ and $0 < \mathfrak{b} < 1$ determine decay rate of the couplings and spatial localization properties of the network. For instance, larger values of $\mathfrak{a}$ and $\mathfrak{b}$ result in more localized networks with short range couplings. To make our study generic, the coefficients ζ_{ij} are independently and randomly chosen from $\mathcal{N}(0,1)$. In our simulations, we set $n = 40$, $\mathfrak{a} = 1$, $\mathfrak{b} = 0.5$, and $d = 0.3$. We generate and save one state matrix $\mathbf{A}$ that has both stable and unstable modes and use it in the following simulation studies. The spatial locations of subsystems and their coupling topology are illustrated in Fig. 5.2.

Constructing Observability Frame: We set $\Omega = \{1, \ldots, n\}$, $\tau = 0.12$, and $M_i = M = 44$ and utilize Corollary 5.6.2 to construct an observability frame Φ with format (5.27). The resulting frame contains $nM = 1760$ vectors with a space-time representation illustrated by blue dots in Fig. 6.2. For a noise intensity of $\sigma = 0.1$, the value of the (least-squares)

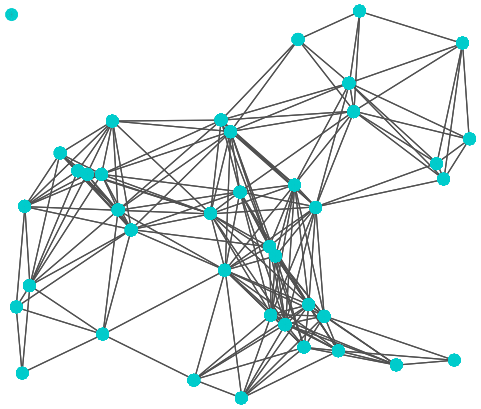

Figure 5.2: The spatial location of subsystems and their coupling structure.

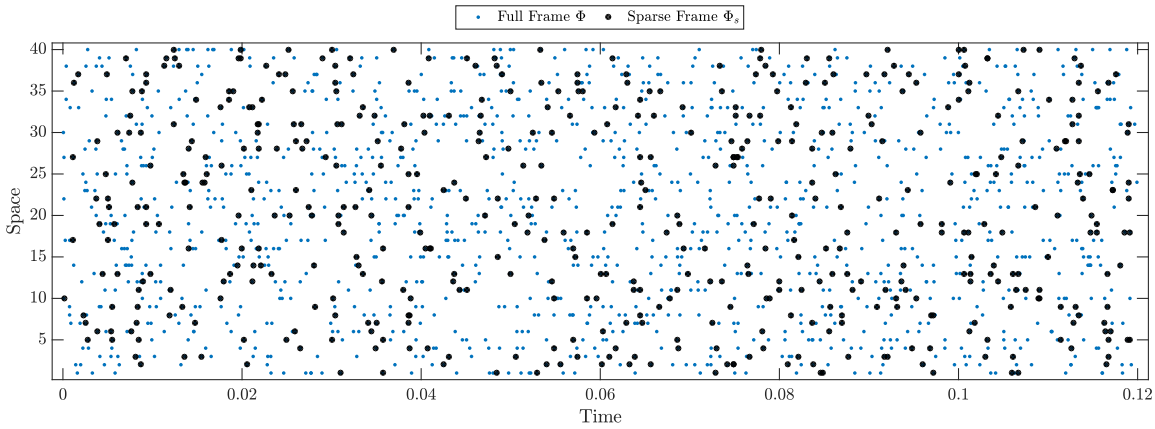

Figure 5.3: The space-time representation of the sampling strategies corresponding to frame Φ and sparsified frame Φ_s, with $|\Phi| = 1760$ and $|\Phi_s| = 503$.

estimation measure is

$$\rho_d(\Phi) \approx 0.0967.$$

Randomized Frame Sparsification: For design parameters $\epsilon = 0.5$ and $q = 590$, we apply Theorem 7.5.2 and find a sparsified frame with $|\Phi_s| = 503$. The spatial locations and time stamps of the sampling strategy corresponding to the sparsified frame are illustrated by black circles in Fig. 6.2. The value of the estimation measure is

$$\rho_d(\Phi_s) \approx 0.1883.$$

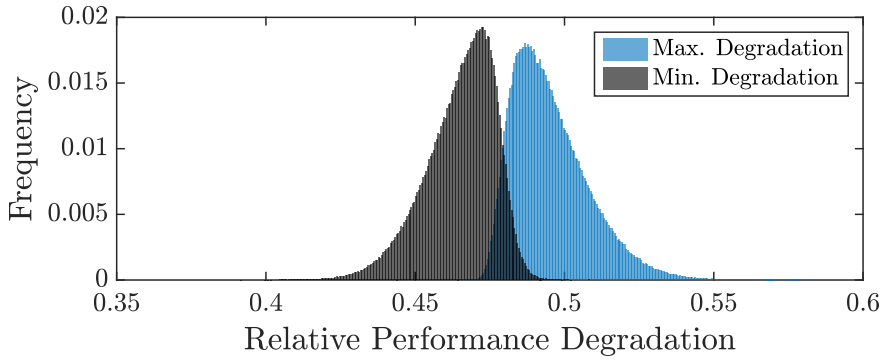

Figure 5.4: The histogram of the performance degradation after random partitioning.

The number of space-time samples has been reduced by 71% for the price of 95% relative estimation-quality loss.

Sparsification via Random Partitioning: Let us reconsider the frame Φ shown in Fig. 6.2. By applying Proposition 5.7.5, random partitioning of Φ leads into two subsets Φ_1 and Φ_2. In our simulations, we repeat the random partitioning procedure 5×10^5 times. In each case, the minimum and maximum relative estimation-quality losses for Φ_1 and Φ_2, i.e.,

$$\max_{j=1,2} \frac{\rho_d(\Phi_j) - \rho_d(\Phi)}{\rho_d(\Phi)} \quad \text{and} \quad \min_{j=1,2} \frac{\rho_d(\Phi_j) - \rho_d(\Phi)}{\rho_d(\Phi)}$$

is computed and saved. The histogram of this data is depicted in Fig. 5.4. In this simulation, the minimum and maximum degradations are less than 0.55 with a high probability. The theoretical estimate from Theorem 5.7.6 is

$$r^* = \max_{\phi \in \Phi} \; r_\phi(\mathbf{S}) \; \approx \; 0.0315 \Rightarrow \kappa(r^*) \approx 1.1441.$$

Our extensive simulations reveal that, in practice, one typically achieves comparably better estimation quality than our theoretical bounds (5.49). In these simulations, the number of space-time samples are reduced by almost 50% for the price of 55% estimation-quality loss (in most outcomes of the simulations).

Performance of Randomized vs. Greedy Algorithms: In this simulation, we compare the estimation quality of the resulting sparsified frames from Algorithm 6 and 5. Using Algo-

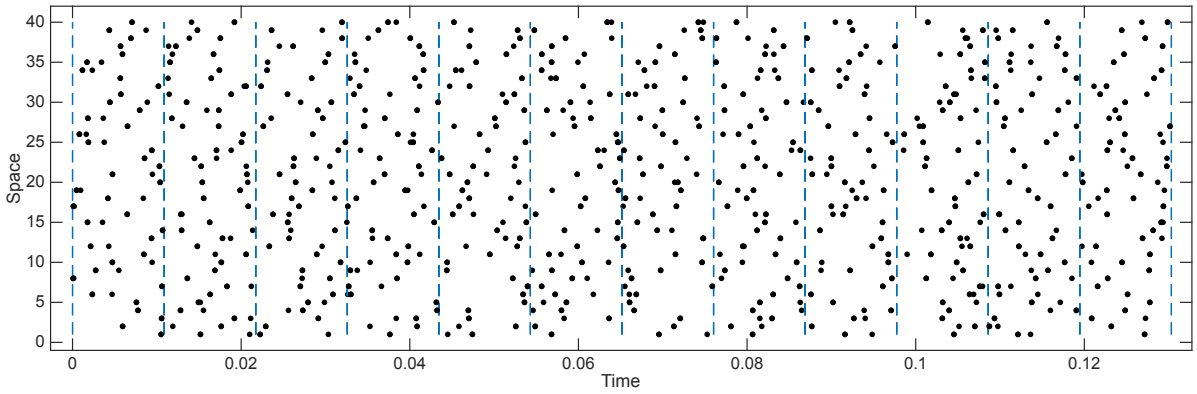

Figure 5.5: The space-time representation of the sampling points of the sequence of frames $\Phi_1, \ldots, \Phi_{12}$ that are separated by the dashed lines.

rithm 6, we construct 25 different sparsified frames by selecting 25 different values for ϵ in $(1/\sqrt{n}, 1]$. We treat q as a control parameter and vary its value between 5902 and 153. For a fixed q, we compute the value of the estimation measure for all 25 frames and save the one with the minimum value. When applying Algorithm 5, we change the desired sparsity level $\mathfrak{s}$ to get a sequence of sparsified frames. The outcome of our simulations is depicted in Fig. 5.6, where one can observe that both methods result in almost similar estimation qualities. The only difference we can report is their running time (on a personal computer with an Intel processor using MATLAB): the randomized method (including 25 experiments per q) took about 1.68 seconds, while the greedy method took 26.71 seconds. This is consistent with our running time analysis for both algorithms.

Sequential Frame Construction: We compute the value of δ^*, which is defined in Theorem 5.6.7, using the saved state matrix $\mathbf{A}$ and get $\delta^* \approx 0.0434$. We merge $N = 12$ subframes in the time horizon, where $c_j = 0.25$ is for every subframe Φ_j for $j = 1, \ldots, N$. Sampling times t_{ij} for each frame is chosen randomly and uniformly from time interval $[t_j, t_{j+1}]$. The space-time representation of these frames is illustrated in Fig. 5.5. The resulting concatenated frame has $|\Phi| = Nn = 480$ components with estimation quality

$$\rho_d(\Phi) \approx 0.1862.$$

Estimation Quality Deterioration with Time Shifts: Let us consider the first observability

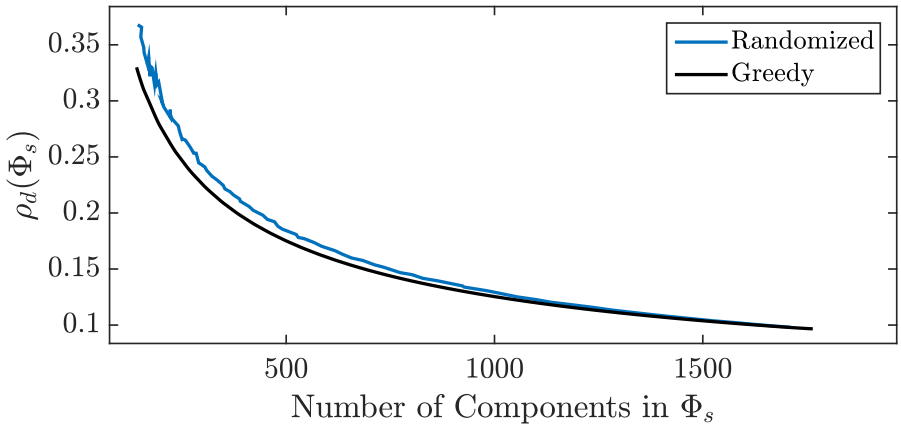

Figure 5.6: The estimation measure of the sparsified frames resulting from Algorithm 6 (the randomized sparsification) and Algorithm 5 (the greedy sparsification) are compared.

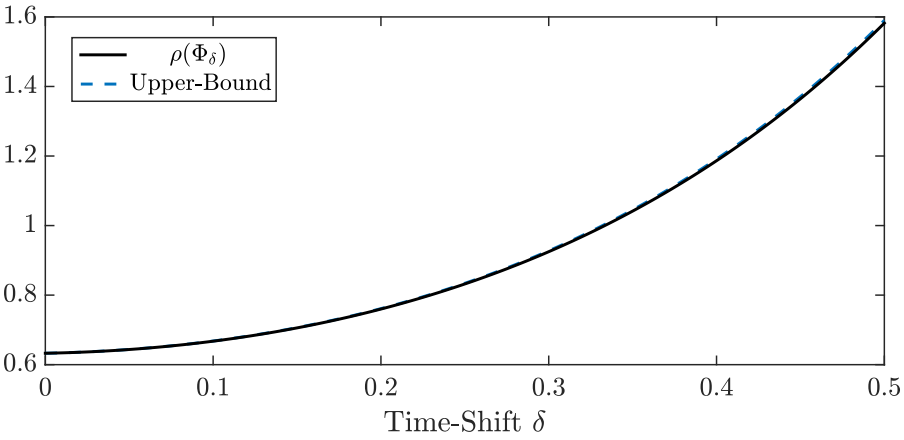

Figure 5.7: The estimation measure of the shifted frames is compared to our theoretical upper bound (5.22).

frame Φ_1 in Fig. 5.5 with n components. First, we construct a family of observability frame Φ_δ, which is defined in (5.21), by increasing δ from 0 to 0.5. We compute exact value of the (least-squares) estimation measure for every Φ_δ. The result of our simulations is depicted in Fig. 5.7 along with our theoretical upper bound (5.22). One observes that our proposed upper bound is rather tight for all values of δ in $[0, 0.5]$.

5.10 Discussion and Conclusion

In this chapter, we assume that measurement noises in (5.14) are Gaussian and independent of each other. When the measurement noises are dependent, the covariance of the estimation error (5.16) will be

$$\mathbf{\Sigma}_\eta = \left(\mathbf{T}^T \mathbf{\Sigma}_\xi^{-1} \mathbf{T}\right)^{-1} \tag{5.69}$$

where $\mathbf{\Sigma}_\xi$ is the covariance of the measurement noise. In general, we may not be able to expand (5.69) as a sum of rank-one matrices made of frame components. This was a useful property for our developments in Section 5.7. This case needs a thorough analysis which is beyond the scope of this Chapter. However, in the case that $\mathbf{\Sigma}_\xi$ is diagonal and with entries

$$\mathbf{\Sigma}_\xi = \mathrm{diag}(\sigma_{i,t}^2)_{(i,t)\in\mathfrak{S}},$$

i.e., the observations are spatially and temporally independent, one can conduct a similar analysis by defining the frame components as

$$\Phi(\mathbf{A}, \mathfrak{S}) \;=\; \left(\sigma_{i,t}^{-1}\, \mathrm{e}^{\mathbf{A}^T t} e_i \;\middle|\; (i,t) \in \mathfrak{S}\right),$$

where $\sigma_{i,t}^2$ is the variance of the observation error at (i,t).

Our results can be extended to include linear dynamical networks with arbitrary output matrices. In such networks, the sampling locations will be different from subsystem locations and the components of the observability frame (5.13) will take form $\mathrm{e}^{\mathbf{A}^T t} c^T$, where c is a row of output matrix. The main reason for working with output matrix (5.4) is to highlight inherent tradeoffs between minimum required number of samples in space and time in order to achieve a certain estimation quality in linear dynamical networks; see the results of Section 5.8.

In order to quantify estimation quality, we consider two estimation measures: standard deviation and differential entropy of the estimation error. Depending on the specific design criteria, one may choose another type of estimation measure. For instance, if the reliability of the estimators is of significant importance for the network designer, one can utilize risk

measures to evaluate the estimation quality. Two useful risk measures are: risk of large aggregate deviations, i.e.,

$$\rho_a(\Phi) := \inf\Big\{\Delta \in \mathbb{R}_+ \;\Big|\; \mathbb{P}\big\{\|\eta\| \geq \Delta\big\} \leq \epsilon\Big\}$$

or risk of large element-wise deviations, i.e.,

$$\rho_r(\Phi) := \inf\Big\{\Delta \in \mathbb{R}_+ \;\Big|\; \mathbb{P}\big\{|\eta_i| \geq \Delta\big\} \leq \epsilon\Big\}.$$

One can show that these two risk measures are monotone according to Definition 5.5.1 and as a result, one can effectively employ these measures instead of ρ_d and ρ_e in our proposed methodology.

The significance of fundamental limits and tradeoffs in Section 5.8 is that they reveal what is achievable and what is not. This is practically plausible as it prevents us from searching for sampling strategies with unachievable estimation qualities.

The results of Section 5.7 provide three methods to sparsify a given observability frame. Our theoretical error bounds are rather conservative. However, our extensive simulations assert that our proposed algorithms can achieve comparably better error bounds in practice.

Appendix I: Proofs

Proof of Theorem 5.4.5: Let us assume that the family of vectors (5.13) is a frame for $\mathbb{R}^n$ with analysis operator $\mathbf{T}$. From Lemma 5.4.4, initial condition can be recovered via

$$x_0 = \mathbf{T}^\dagger y, \tag{5.70}$$

where $y = [\langle x, \phi_i\rangle]_{\phi_i \in \Phi}$. In the next step, suppose that (5.13) is not a frame for $\mathbb{R}^n$. Then, the frame matrix $\mathbf{S}$ will be singular by Proposition 5.4.3. Thus, for every $x_1 \in \mathbb{R}^n$, two initial states x_1 and $x_1 + x_2$, in which x_2 is a nonzero element of null($\mathbf{S}$), produce the same vector of observation y. This is contrary to our assumption on unique determination of the initial state of the network.

Proof of Lemma 5.4.6: Based on the Cayley-Hamilton theorem

$$\mathrm{e}^{\mathbf{A}t} = \sum_{k=0}^{n-1} g_k(t)\mathbf{A}^k, \tag{5.71}$$

for some functions $g_k(t)$ for $k = 0, \ldots, n-1$. Using this fact, the analysis matrix corresponding to the sampling strategy $\mathfrak{S}$ can be written as

$$\mathbf{T} = \left[\sum_{k=0}^{n-1} g_k(t)\mathbf{A}^k e_i\right]_{(i,t)\in\mathfrak{S}}. \tag{5.72}$$

Assume that Φ is a frame but the pair $(\mathbf{A}, \mathbf{C}_\Omega)$ is not observable. Thus, the observability matrix

$$O(\mathbf{A}, \mathbf{C}_\Omega) = \left[\left(\mathbf{A}^k \mathbf{C}_\Omega^T\right)^T\right]_{k=0,\ldots,n-1} \tag{5.73}$$

has rank less than n. Because Φ is a frame, rank of $\mathbf{T}$ is n. However, comparing (5.72) with the form of the observability matrix, we observe that rank of matrix $\mathbf{T}$ in (5.72) is at most equal to rank of $O(\mathbf{A}, \mathbf{C}_\Omega)$. This is a contradiction, proving that $(\mathbf{A}, \mathbf{C}_\Omega)$ must be observable.

Proof of Proposition 5.5.2: The expression for the estimation measure holds because we can write

$$\begin{aligned}\rho_d(\Phi)^2 &= \mathbb{E}\{\|\eta\|_2^2\} = \mathbb{E}\Big\{\xi^T \mathbf{T}\mathbf{S}^{-2}\mathbf{T}^T\xi\Big\} = \mathbb{E}\Big\{\mathrm{Tr}\big(\mathbf{S}^{-2}\mathbf{T}^T\xi\xi^T\mathbf{T}\big)\Big\} \\ &= \sigma^2\mathrm{Tr}(\mathbf{S}^{-1}) = \sigma^2\sum_{i=1}^{n}\frac{1}{\lambda_i(\mathbf{S})}.\end{aligned}$$

To see that ρ_d is a monotone operator, consider the following chain of observations:

$$\Phi_1 \subseteq \Phi_2 \Rightarrow \mathbf{S}_1 \preceq \mathbf{S}_2 \Rightarrow \mathbf{S}_2^{-1} \preceq \mathbf{S}_1^{-1} \Rightarrow \mathrm{Tr}(\mathbf{S}_2^{-1}) \leq \mathrm{Tr}(\mathbf{S}_1^{-1}).$$

The last inequality implies that $\rho_d(\Phi_2) \leq \rho_d(\Phi_1)$.

Proof of Proposition 5.5.3: For the random variable η, using (5.19), we can show that (see

Chapter 8 in [146])

$$\begin{aligned} h(\eta) &= \frac{1}{2}\log\big(\det(\sigma^2\mathbf{S}^{-1})\big) + \frac{n}{2}\log(2\pi e) \\ &= \frac{1}{2}\sum_{i=1}^{n}\log\left(\frac{\sigma^2}{\lambda_i(\mathbf{S})}\right) + \frac{n}{2}\log(2\pi e) \\ &= \frac{1}{2}\rho_e(\Phi) + \frac{n}{2}\big(1+\log(2\pi\sigma^2)\big). \end{aligned} \tag{5.74}$$

To see that ρ_e is a monotone operator, consider the following chain of operations:

$$\begin{aligned} \Phi_1 \subseteq \Phi_2 \quad &\Rightarrow \quad \mathbf{S}_1 \preceq \mathbf{S}_2 \\ &\Rightarrow \quad \mathbf{S}_2^{-1} \preceq \mathbf{S}_1^{-1} \\ &\Rightarrow \quad \log(\mathbf{S}_2^{-1}) \preceq \log(\mathbf{S}_1^{-1}) \\ &\Rightarrow \quad \mathrm{Tr}(\log(\mathbf{S}_2^{-1})) \leq \mathrm{Tr}(\log(\mathbf{S}_1^{-1})). \end{aligned}$$

The third inequality holds because $\log(.)$, as a map from the cone of positive definite matrices to the set of symmetric matrices, is analytic and increasing on the cone of positive definite matrices. The last inequality implies that $\rho_e(\Phi_2) \leq \rho_e(\Phi_1)$.

Proof of Proposition 5.5.4: Let us denote the analysis matrices corresponding to Φ and Φ_δ by $\mathbf{T}$ and $\mathbf{T}_\delta$, respectively. It follows that

$$\mathbf{T}_\delta = \mathbf{T}\mathrm{e}^{\mathbf{A}\delta}.$$

Thus, the corresponding frame matrix for Φ_δ is

$$\mathbf{S}_\delta = \mathbf{T}_\delta^T\mathbf{T}_\delta = \mathrm{e}^{\mathbf{A}^T\delta}\mathbf{T}^T\mathbf{T}\mathrm{e}^{\mathbf{A}\delta} = \mathrm{e}^{\mathbf{A}^T\delta}\mathbf{S}\mathrm{e}^{\mathbf{A}\delta},$$

where $\mathbf{S}$ is the frame matrix of Φ. As a result, we can see that

$$\mathbf{S}_\delta^{-1} = \mathrm{e}^{-\mathbf{A}\delta}\mathbf{S}^{-1}\mathrm{e}^{-\mathbf{A}^T\delta}.$$

Therefore, estimation using the shifted frame Φ_δ results in a normal error (random) variable

$$\eta \sim \mathcal{N}\left(0, \sigma^2 \mathbf{S}_\delta^{-1}\right) = \mathcal{N}\left(0, \sigma^2 \mathrm{e}^{-\mathbf{A}\delta} \mathbf{S}^{-1} \mathrm{e}^{-\mathbf{A}^T\delta}\right). \tag{5.75}$$

In the next step, the estimation measures can be found as follows

$$\begin{aligned}
\frac{1}{\sigma^2}\,\rho_d(\Phi_\delta)^2 &= \mathrm{Tr}\left(\mathrm{e}^{-\mathbf{A}\delta}\mathbf{S}^{-1}\mathrm{e}^{-\mathbf{A}^T\delta}\right) \\
&= \mathrm{Tr}\left(\mathrm{e}^{-\mathbf{A}^T\delta}\mathrm{e}^{-\mathbf{A}\delta}\mathbf{S}^{-1}\right) \\
&= \mathrm{Tr}\left(\left(\mathrm{e}^{\mathbf{A}\delta}\mathrm{e}^{\mathbf{A}^T\delta}\right)^{-1}\mathbf{S}^{-1}\right) \\
&\leq \sum_{i=1}^{n} \sigma_i\left(\left(\mathrm{e}^{\mathbf{A}\delta}\mathrm{e}^{\mathbf{A}^T\delta}\right)^{-1}\right)\sigma_i(\mathbf{S}^{-1}),
\end{aligned}$$

where in the last equality we have used Von Neumann's trace inequality; we refer to [147] for more details. We can further write

$$\begin{aligned}
\rho_d(\Phi_\delta)^2 &\leq \sigma^2 \sum_{i=1}^{n} \sigma_i^2\left(\mathrm{e}^{-\mathbf{A}\delta}\right)\lambda_i(\mathbf{S}^{-1}) \\
&= \sigma^2 \sum_{i=1}^{n} \frac{1}{\sigma_i^2(\mathrm{e}^{\mathbf{A}\delta})\cdot\lambda_i(\mathbf{S})}.
\end{aligned}$$

For the last part of the proof, we have

$$\begin{aligned}
\rho_e(\Phi_\delta) &= -\log(\det(\mathbf{S}_\delta)) = -\log\left(\det(\mathrm{e}^{\mathbf{A}^T\delta}\mathbf{S}\mathrm{e}^{\mathbf{A}\delta})\right) \\
&= -\log\left(\det(\mathrm{e}^{\mathbf{A}^T\delta})\right)\det\left((\mathrm{e}^{\mathbf{A}\delta})\right) - \log\det(\mathbf{S}) \\
&= \rho_e(\Phi) - \sum_{i=1}^{n}\log\left(\lambda_i(\mathrm{e}^{\mathbf{A}^T\delta}\mathrm{e}^{\mathbf{A}\delta})\right) \\
&= \rho_e(\Phi) - \sum_{i=1}^{n}\log\left(\sigma_i^2(\mathrm{e}^{\mathbf{A}\delta})\right).
\end{aligned}$$

Proof of Theorem 5.6.1: When $\mathbf{A} = \mathbf{0}$, the proof becomes trivial as in this case $\mathrm{e}^{\mathbf{A}t} = \mathbf{I}$ for all $t \in \mathbb{R}$ and $\mathbf{C}_\Omega$ must be the identity matrix according to Assumption 5.3.1. When $\mathbf{A}$ is

nonzero, its minimal polynomial has degree $\mathfrak{d}(\mathbf{A})$ with $1 \leq \mathfrak{d}(\mathbf{A}) \leq n$. Then, it follows that

$$\mathrm{e}^{\mathbf{A}t} = \sum_{k=0}^{\infty} \frac{t^k}{k!} \mathbf{A}^k = \sum_{k=0}^{\mathfrak{d}(\mathbf{A})-1} g_k(t) \mathbf{A}^k \tag{5.76}$$

for all $t \in \mathbb{R}$, where g_k's are some functions of time. Let $\mathbf{J}$ be the Jordan canonical form of $\mathbf{A}$ and

$$\mathbf{A} = \mathbf{Q}^{-1}\mathbf{J}\mathbf{Q}. \tag{5.77}$$

for some nonsingular matrix $\mathbf{Q}$. By combining (5.76) and (5.77), one obtains

$$\mathrm{e}^{\mathbf{J}t} = \sum_{k=0}^{\mathfrak{d}(\mathbf{A})-1} g_k(t) \mathbf{J}^k \tag{5.78}$$

for all $t \in \mathbb{R}$. Let us consider the minimal polynomial of $\mathbf{A}$ given by (5.24), where p_m's are some integer numbers for $1 \leq m \leq q$. From the definition of a minimal polynomial, there are Jordan blocks $\mathbf{J}_{\lambda_m,p_m}$ associated with every eigenvalue $\lambda_m(\mathbf{A})$ whose dimension is p_m. This together with (5.78) implies that

$$\mathrm{e}^{\mathbf{J}_{\lambda_m,p_m}t} = \sum_{k=0}^{\mathfrak{d}(\mathbf{A})-1} g_k(t) \left(\mathbf{J}_{\lambda_m,p_m}\right)^k,$$

or equivalently,

$$\mathrm{e}^{\lambda_m(\mathbf{A})t} \frac{t^l}{l!} = \sum_{k=l}^{\mathfrak{d}(\mathbf{A})-1} g_k(t) \binom{k}{l} \lambda_m(\mathbf{A})^{k-l} \tag{5.79}$$

for all $0 \leq l \leq p_j - 1$ and $t \in \mathbb{R}$. The last equivalence holds as (j, j')'th entry of $(\mathbf{J}_{\lambda_m,p_m})^k$ with property $0 \leq j' - j \leq \min(k, p_m - 1)$ equals to $\binom{k}{j'-j}\lambda_m(\mathbf{A})^{k-j+j'}$ and all other entries are equal to zero. It is well known that functions $\mathrm{e}^{\lambda_m(\mathbf{A})t}t^k$ for $m = 1, \ldots, q$ and $k = 0, \ldots, p_m - 1$ for $t \in \mathbb{R}$ are linearly independent. This, together with (5.79), and the fact that

$$\mathfrak{d}(\mathbf{A}) = \sum_{m=1}^{q} p_m$$

implies the existence of a nonsingular matrix $\mathbf{D}$ that satisfies

$$G(t) = \mathbf{D}E(t), \tag{5.80}$$

where $E(t)$ is defined in (5.25) and

$$G(t) = \big[g_k(t)\big]_{0\leq k\leq \mathfrak{d}(\mathbf{A})-1}.$$

Let us denote $\Theta_i = \big\{t_{i,j} \;\big|\; 1 \leq j \leq M_i\big\}$. According to our assumptions, matrix $\mathbf{E_i}$ defined by (5.26) has full row rank $\mathfrak{d}(\mathbf{A})$. This, together with (5.76) and (5.80), implies the existence of scalars $a_{i,j,k}$ such that

$$\mathbf{A}^k \;=\; \sum_{j=1}^{M_i} a_{i,j,k}\; \mathrm{e}^{\mathbf{A}^T t_{i,j}} \tag{5.81}$$

for $0 \leq k \leq \mathfrak{d}(\mathbf{A}) - 1$. Let us denote

$$\mathbf{C}_\Omega = \left[\; e_{i_1} \;\middle|\; \ldots \;\middle|\; e_{i_p} \;\right]^T$$

and

$$\mathbf{F} = \Big[(\mathbf{A}^T)^k\, e_i\Big]_{\substack{i\in\Omega \\ 0\leq k\leq \mathfrak{d}(\mathbf{A})-1}}.$$

Based on Assumption 5.3.1, and the fact that $\mathfrak{d}(\mathbf{A})$ is greater than or equal to the observability index of $(\mathbf{A}, \mathbf{C})$, for $n \times \mathfrak{d}(\mathbf{A})|\Omega|$ matrix $\mathbf{F}$ it holds that

$$\mathrm{rank}(\mathbf{F}) = n, \tag{5.82}$$

(for example see Section 6.3.1 in [148]). Let us define

$$\mathbf{T} = \Big[\mathrm{e}^{\mathbf{A}^T t}\, e_i\Big]^T_{i\,\in\,\Omega,\; t\,\in\,\Theta_i}.$$

From (5.81), it follows that the rank of matrix $\mathbf{T}$, whose size is $n \times |\mathfrak{S}|$, is larger than or equal to rank of $\big[\mathbf{C}_\Omega \mathbf{A}^k\big]_{0\leq k\leq \mathfrak{d}(\mathbf{A})-1}$. This together with (5.82) implies that $\mathbf{T}$ has rank n, which implies that the family of vectors (5.27) form a frame for $\mathbb{R}^n$.

Proof of Corollary 5.6.2: First suppose that $|\Theta_i| = \mathfrak{d}(\mathbf{A})$ for each $i \in \Omega$. Observe that nonzero combinations of $\mathrm{e}^{\lambda_i(\mathbf{A})t} t^k$ for $i = 1, \ldots, q$ and $k = 0, \ldots, p_i - 1$ have only finitely

many zeros. This shows that for any $\tau > 0$ and $i \in \Omega$

$$\det(\mathbf{E}_i) = \det([E(t_j)]_{j=1,\ldots,\mathfrak{d}(\mathbf{A})}) \neq 0,$$

for almost every choice of times $[t_1, \ldots, t_{\mathfrak{d}(\mathbf{A})}] \in [0,\tau]^{\mathfrak{d}(\mathbf{A})}$. Hence, if we choose the sampling times $t_j \in \Theta_i$ with $|\Theta_i| = \mathfrak{d}(\mathbf{A})$ randomly in the range $[0,\tau]$ in an independent and uniform manner, then with probability one the requirement (5.26) is satisfied and $\mathbf{E}_i$ is full rank for each $i \in \Omega$. Thus, by Theorem 5.6.1, the resulting family of vectors is a frame with probability one. If the number of random samples per location $i \in \Omega$ increases beyond $\mathfrak{d}(\mathbf{A})$, the result is still a frame.

Proof of Theorem 5.6.4: First assume that for each location $i \in \Omega$, we choose $|\Theta_i| = \mathfrak{d}(\mathbf{A})$. In this case, for every location $i \in \Omega$

$$\mathbf{E}_i = [E(k\delta)]_{k=0,\ldots,\mathfrak{d}(\mathbf{A})-1}.$$

one observes that

$$\det(\mathbf{E}_i) = C \prod_{1 \leq m < m' \leq q} \left(\mathrm{e}^{\lambda_m(\mathbf{A})\delta} - \mathrm{e}^{\lambda_{m'}(\mathbf{A})\delta} \right)^{p_m p_{m'}}$$

for some nonzero number C depending only on p_m with $1 \leq m \leq q$. So the requirement (5.26) is satisfied if

$$(\lambda_m(\mathbf{A}) - \lambda_{m'}(\mathbf{A}))\delta \notin 2\pi \mathrm{j}\, \mathbb{Z}$$

for all distinct eigenvalues $\lambda_m(\mathbf{A})$ and $\lambda_{m'}(\mathbf{A})$. Therefore by Theorem 5.6.1, for $\mathbf{B}_\delta := \mathrm{e}^{\mathbf{A}^T\delta}$, we know that

$$\Phi' := \left(\mathbf{B}_\delta^k e_i \middle| \ i \in \Omega,\ k = 0, \ldots, \mathfrak{d}(\mathbf{A}) - 1 \right). \tag{5.83}$$

is a frame. The family of vectors Φ given in the theorem satisfies $\Phi' \subseteq \Phi$. Thus, Φ is also a frame.

Proof of Theorem 5.6.7: Since matrix $\mathrm{e}^{\mathbf{A}^T t^*}$ is full rank for all $t^* \geq 0$, one can verify that Φ in (5.35) forms a frame for $\mathbb{R}^n$ if and only if $\left(\mathrm{e}^{\mathbf{A}^T(t-t^*)} e_i \mid (i,t) \in \mathfrak{S} \right)$ forms a frame. Due to

this shift-invariance property, we may safely assume that $t^* = 0$ by shifting every element in all Θ_i's by $-t^*$. Let us pick $t_i \in \Theta_i$ for $1 \leq i \leq n$. Then, for every vector $c = [c_1, \ldots, c_n]^T$, we have

$$\begin{aligned}
\left\| \sum_{i=1}^{n} c_i e^{\mathbf{A}t_i} e_i - \sum_{i=1}^{n} c_i e_i \right\| &= \left\| \sum_{m=1}^{\infty} \frac{\mathbf{A}^m}{m!} \sum_{i=1}^{n} c_i t_i^m e_i \right\| \\
&\leq \sum_{m=1}^{\infty} \frac{\|\mathbf{A}\|^m}{m!} \left\| \sum_{i=1}^{n} c_i t_i^m e_i \right\| \\
&\leq \sum_{m=1}^{\infty} \frac{\|\mathbf{A}\|^m (\delta^*)^m}{m!} \|c\| \\
&= \left(e^{\|\mathbf{A}\|\delta^*} - 1 \right) \|c\|.
\end{aligned}$$

This implies that

$$\left(2 - e^{\|\mathbf{A}\|\delta^*} \right) \|c\| \leq \left\| \sum_{i=1}^{n} c_i e^{\mathbf{A}t_i} e_i \right\| \leq e^{\|\mathbf{A}\|\delta^*} \|c\|$$

for all $c \in \mathbb{R}^n$. Hence, Φ, with $|\Theta_i| = 1$ for every sampling location $i \in \Omega$, consists of n linearly independent vectors and forms a frame for $\mathbb{R}^n$. By increasing the number of samples per sampling location, Φ will remain to be a frame for $\mathbb{R}^n$.

Proof of Corollary 5.6.8: First, for each $i \in \Omega$, consider $|\Theta_i| = 1$ and denote $\Theta_i = \{t_i\}$. Moreover, we define

$$\theta = [t_i]_{i=1,\ldots,n}. \tag{5.84}$$

Now, we denote $\mathbf{S}(\theta)$ to be the frame matrix corresponding to family of vectors (5.35), where the sampling times have been gathered in $\theta \in \mathbb{R}^n$. Based on Theorem 5.6.7, there exist a θ for which the real analytic function $F(\theta) : \mathbb{R}^n \to \mathbb{R}$ defined by

$$F(\theta) := \det(\mathbf{S}(\theta)),$$

is nonzero. Thus, measure of points $\theta \in [0, \tau]^n$ for which $F(\theta)$ vanishes is zero (e.g. see [149]), and independent and uniform sampling of the time stamps in $[0, \tau]^n$ gives a frame with

probability one. If we increase Θ_i beyond 1 probability of getting a frame is 1.

Proof of Theorem 5.7.3: We build up our proof based on some of the steps taken in the proof of Theorem 5 in [57]. Using the leverage scores (5.41), one can verify that

$$\mathbf{0} \prec (1-\epsilon)\,\mathbf{S} \preceq \mathbf{S}_w \tag{5.85}$$

with probability at least 1/2, where

$$\mathbf{S}_w = \sum_{\phi\in\Phi} w_s(\phi)\,\phi\phi^T = \sum_{\phi\in\Phi} \frac{f_\phi}{q\pi(\phi)}\,\phi\phi^T.$$

and $f_\phi \geq 0$ is the frequency of the times that ϕ is sampled (due to sampling with replacement, some vectors can be sampled multiple times). Weights of those $\phi \notin \Phi_s$ are equal to 0. For an outcome of Algorithm 6, one has

$$\mathbf{0} \prec \mathbf{S}_w \preceq \left(\max_{\phi\in\Phi_s} w_s(\phi)\right) \sum_{\phi\in\Phi_s} \phi\phi^T$$

This implies that $\mathbf{S}_s = \sum_{\phi\in\Phi_s} \phi\phi^T \succ \mathbf{0}$ with probability at least 1/2. For the first part of our proof, Φ_s is a frame for $\mathbb{R}^n$ if and only if $\mathbf{S}_s \succ \mathbf{0}$.

Denote the analysis operator of Φ_s by $\mathbf{T}_s$ and set $\mathbf{W}_s = \mathrm{diag}\big(w_s(\phi)\big)\big|_{\phi\in\Phi_s}$. Let

$$y_s = \mathbf{T}_s x_0 + \xi_s \tag{5.86}$$

be a noisy observation vector collected by $\phi \in \Phi_s$, in which $\xi_s \in \mathbb{R}^m$ is a zero mean Gaussian measurement noise with independent components and covariance $\mathbb{E}\left\{\xi\xi^T\right\} = \sigma^2\mathbf{I}$. In the next step, let us consider an alternative estimator $\tilde{x}_0$ that is given by

$$\tilde{x}_0 \;=\; \left(\mathbf{T}_s^T\mathbf{W}_s\,\mathbf{T}_s\right)^{-1}\mathbf{T}_s^T\mathbf{W}_s y_s.$$

It is straightforward to verify that this is an unbiased estimator using only the observations corresponding to Φ_s. Since covariance of noise is $\mathbb{E}\left\{\xi\xi^T\right\} = \sigma^2\mathbf{I}$, the unweighted least-

squares estimator gives the optimal estimator $\hat{x}_0$. Therefore,

$$\mathbb{E}\left\{\tilde{x}_0\tilde{x}_0^T\right\} \succeq \mathbb{E}\left\{\hat{x}_0\hat{x}_0^T\right\}.$$

This lets us write

$$\begin{aligned}
\rho_d(\Phi_s)^2 &= \sigma^2 \operatorname{Tr}\left(\mathbb{E}\{\hat{x}_0\hat{x}_0^T\}\right) \\
&\leq \sigma^2 \operatorname{Tr}\left(\mathbb{E}\{\tilde{x}_0\tilde{x}_0^T\}\right) \\
&= \sigma^2 \operatorname{Tr}\left(\mathbf{S}_w^{-1}\mathbf{T}^T\mathbf{W}_s^2\mathbf{T}\mathbf{S}_w^{-1}\right) \\
&= \sigma^2 \operatorname{Tr}\left(\mathbf{S}_w^{-1}\hat{\mathbf{S}}_w\mathbf{S}_w^{-1}\right) \qquad (5.87)
\end{aligned}$$

where $\mathbf{S}_w = \mathbf{T}_s^T\mathbf{W}_s\mathbf{T}_s$ and $\hat{\mathbf{S}}_w := \mathbf{T}_s^T\mathbf{W}_s^2\mathbf{T}_s$. From (5.87) and the definition of random variable χ in (5.45), it follows that

$$\rho_d(\Phi_s)^2 \;\leq\; \sigma^2\chi \operatorname{Tr}\big(\mathbf{S}_w^{-1}\mathbf{S}_w\mathbf{S}_w^{-1}\big) \qquad (5.88)$$

in which the middle matrix is replaced by its upper bound as one can show that for every three positive-definite matrices $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ with $\mathbf{X}_1 \succeq \mathbf{X}_2$, inequality $\mathbf{X_3}\mathbf{X_1}\mathbf{X_3} \succeq \mathbf{X_3}\mathbf{X_2}\mathbf{X_3}$ holds. According to Markov inequality, the next inequality holds

$$\chi \leq \frac{1}{1-\frac{3}{4}}\,\mathbb{E}\{\chi\} \;=\; 4\bar{\chi} \qquad (5.89)$$

with probability at least 3/4. From (5.88) and (5.89), we get

$$\rho_d(\Phi_s)^2 \;\leq\; 4\sigma^2\,\bar{\chi}\,\operatorname{Tr}\big(\mathbf{S}_w^{-1}\big). \qquad (5.90)$$

On the other hand, by applying similar steps to the proof of Theorem 5 in [57], it follows that with probability at least 1/2 we have

$$(1-\epsilon)\mathbf{S} \preceq \mathbf{S}_w \preceq (1+\epsilon)\mathbf{S}. \qquad (5.91)$$

By taking inverse, we get

$$(1+\epsilon)^{-1}\mathbf{S}^{-1} \preceq \mathbf{S}_w^{-1} \preceq (1-\epsilon)^{-1}\mathbf{S}^{-1}. \tag{5.92}$$

If the event described in (5.89) is denoted by $\mathfrak{A}$ and the event described by (5.92) is denoted by $\mathfrak{B}$, then

$$\mathbb{P}(\mathfrak{A} \cap \mathfrak{B}) = \mathbb{P}(\mathfrak{A}) + \mathbb{P}(\mathfrak{B}) - \mathbb{P}(\mathfrak{A} \cup \mathfrak{B}) \geq \frac{3}{4} + \frac{1}{2} - 1 = \frac{1}{4}.$$

Therefore both (5.90) and (5.92) hold with probability at least 1/4. Combining (5.90) and (5.92), one arrives at

$$\rho_d(\Phi_s)^2 \leq \frac{4\sigma^2\bar{\chi}}{1-\epsilon}\,\mathrm{Tr}(\mathbf{S}^{-1}) = \frac{4\bar{\chi}}{1-\epsilon}\,\rho_d(\Phi)^2. \tag{5.93}$$

Taking the square root from both sides, we get the desired inequality.

For the entropy estimation measure, by following almost identical steps, we can show that with probability at least 1/4 the following inequality holds

$$\begin{aligned}
\rho_e(\Phi_s) &= \det(\log(\mathbf{S}_s^{-1})) \\
&\leq \log\left(\det\left(\frac{4\bar{\chi}}{1-\epsilon}\mathbf{S}^{-1}\right)\right) \\
&= n\log\left(\frac{4\bar{\chi}}{1-\epsilon}\right) + \rho_e(\Phi).
\end{aligned}$$

Proof of Corollary 5.7.4: We can write

$$\begin{aligned}
\sum_{\phi\in\Phi} w_s(\phi)^2\,\phi\phi^T &\leq \sum_{\phi\in\Phi} w_s(\phi)\Big(\max_{j=1,\ldots,|\Phi|} w_s(\phi)\Big)\,\phi\phi^T \\
&= \Big(\max_{j=1,\ldots,|\Phi|} w_s(\phi)\Big)\sum_{\phi\in\Phi} w_s(\phi)\,\phi\phi^T.
\end{aligned}$$

Therefore, $\max_{\phi\in\Phi} w_s(\phi)$ is an upper-bound on χ. The rest of the proof follows from the fact that we can replace χ with $\max_{\phi\in\Phi} w_s(\phi)$ in the proof of Theorem 5.7.3.

Proof of Theorem 5.7.6: Let us denote $\mathbf{S}_j$ to be the frame matrix corresponding to family of vectors Φ_j that is resulted from the random partitioning according to Proposition 5.7.5.

For the least-squares estimation measure, we have

$$\begin{aligned}\rho_d(\Phi_j) &= \sigma\sqrt{\mathrm{Tr}\big(\mathbf{S}_j^{-1}\big)} \leq \frac{\sigma}{\sqrt{1-\frac{(1+\sqrt{2r})^2}{2}}}\sqrt{\mathrm{Tr}\left(\mathbf{S}^{-1}\right)} \\ &= \Big(1-\frac{(1+\sqrt{2r})^2}{2}\Big)^{-1/2}\rho_d(\Phi).\end{aligned}$$

This proves the first bound.

For the entropy estimation measure, it follows that

$$\begin{aligned}\rho_e(\Phi_j) &= \det(\log(\mathbf{S}_j^{-1})) \\ &\leq \log\left(\det\left((1-\frac{(1+\sqrt{2r})^2}{2})^{-1}\mathbf{S}^{-1}\right)\right) \\ &= -n\log\left(1-\frac{(1+\sqrt{2r})^2}{2}\right)+\rho_e(\Phi),\end{aligned}$$

which proves the second bound.

Proof of Proposition 5.7.7: Taking trace from both sides of (5.53), we get

$$\rho_d^2(\Phi_{i+1}) = \rho_d^2(\Phi_i) + \frac{\sigma^2}{1-\phi^T\mathbf{S}_i^{-1}\phi}\mathrm{Tr}\left(\mathbf{S}_i^{-1}\phi\phi^T\mathbf{S}_i^{-1}\right).$$

Update rule (5.54) follows by utilizing the following equation

$$\mathrm{Tr}\left(\mathbf{S}_i^{-1}\phi\phi^T\mathbf{S}_i^{-1}\right) = \left\|\mathbf{S}_i^{-1}\phi\right\|^2.$$

The update rule (5.55) for the entropy estimation measure follows from

$$\begin{aligned}
\rho_e(\Phi_{i+1}) &= \log\left(\det\left(\mathbf{S}_{i+1}^{-1}\right)\right) \\
&= \log\left(\det\left(\mathbf{S}_i^{-1} + \frac{\mathbf{S}_i^{-1}\phi\phi^T\mathbf{S}_i^{-1}}{1-\phi^T\mathbf{S}_i^{-1}\phi}\right)\right) \\
&= \log\left(\left(1 + \frac{(\mathbf{S}_i^{-1}\phi)^T(\mathbf{S}_i^{-1})^{-1}\mathbf{S}_i^{-1}\phi}{1-\phi^T\mathbf{S}_i^{-1}\phi}\right)\det(\mathbf{S}_i^{-1})\right) \\
&= \log\left(\left(1 + \frac{\phi^T\mathbf{S}_i^{-1}\phi}{1-\phi^T\mathbf{S}_i^{-1}\phi}\right)\det(\mathbf{S}_i^{-1})\right) \\
&= \log\left(\left(\frac{1}{1-\phi^T\mathbf{S}_i^{-1}\phi}\right)\det(\mathbf{S}_i^{-1})\right) \\
&= \log(\det(\mathbf{S}_i^{-1})) + \log\left(\frac{1}{1-\phi^T\mathbf{S}_i^{-1}\phi}\right) \\
&= \rho_e(\Phi_i) - \log(1-\phi^T\mathbf{S}_i^{-1}\phi).
\end{aligned}$$

In the third line, the matrix determinant lemma is applied [137].

Proof of Theorem 5.8.1: First, we prove a more general inequality. Let us consider the class of all estimation measures that have the following spectral representation

$$\rho(\Phi) = \sum_{i=1}^{n} \psi(\lambda_i(\mathbf{S})), \tag{5.94}$$

for some convex and monotonically decreasing function $\psi : \mathbb{R}_+ \to \mathbb{R}$. Since ψ is convex, we apply Jensen's inequality [150] and write

$$\begin{aligned}
\frac{1}{n}\rho(\Phi) &= \frac{1}{n}\sum_{i=1}^{n}\psi(\lambda_i(\mathbf{S})) \\
&\geq \psi\left(\sum_{i=1}^{n}\frac{\lambda_i(\mathbf{S})}{n}\right) = \psi\left(\frac{\mathrm{Tr}(\mathbf{S})}{n}\right).
\end{aligned} \tag{5.95}$$

On the other hand, we have

$$\begin{aligned}
\mathrm{Tr}\left(\mathbf{S}\right) &= \mathrm{Tr}\left(\sum_{i=1}^{|\mathfrak{S}|}\phi_i\phi_i^T\right) = \sum_{i=1}^{|\mathfrak{S}|}\mathrm{Tr}\left(\phi_i\phi_i^T\right) \\
&= \sum_{i=1}^{|\mathfrak{S}|}\phi_i^T\phi_i = \sum_{i=1}^{|\mathfrak{S}|}\|\phi_i\|^2.
\end{aligned} \tag{5.96}$$

Each vector ϕ_i corresponds to some time stamp t and some index $j \in \Omega$, i.e.,

$$\phi_i = \mathrm{e}^{\mathbf{A}^T t} e_j.$$

Thus, we use the bound on the matrix exponential to get

$$\|\phi_i\| = \|\mathrm{e}^{\mathbf{A}^T t} e_j\| \leq \|\mathrm{e}^{\mathbf{A}^T t}\| \, \|e_j\| \leq \nu(\mathbf{A}, \mathfrak{S}).$$

This inequality together with (5.96) gives us

$$\mathrm{Tr}\,(\mathbf{S}) \leq \sum_{i=1}^{|\mathfrak{S}|} \|\phi_i\|^2 \leq \sum_{i=1}^{|\mathfrak{S}|} \nu(\mathbf{A}, \mathfrak{S})^2 = \nu(\mathbf{A}, \mathfrak{S})^2 \, |\mathfrak{S}|.$$

Since ψ is monotonically decreasing, from (5.95), one may conclude that

$$\frac{1}{n}\rho(\Phi) \geq \psi\left(\frac{\mathrm{Tr}(\mathbf{S})}{n}\right) \geq \psi\left(\frac{\nu(\mathbf{A}, \mathfrak{S})^2 \, |\mathfrak{S}|}{n}\right). \tag{5.97}$$

By applying inequality (5.97) to spectral functions $\psi(\lambda) = \lambda^{-1}$ and $\psi(\lambda) = -\log(\lambda)$, we will get the desired inequalities (5.58) and (5.59), respectively.

Proof of Theorem 5.8.2: Every sampling strategy $\mathfrak{S}$ that satisfies our assumptions also satisfies

$$\mathfrak{S} \subset \Omega \times \delta\mathbb{Z}.$$

Therefore, the frame matrix corresponding to such sampling strategy satisfies

$$\mathbf{S} \preceq \sum_{i \in \Omega} \sum_{t \in \delta\mathbb{Z}_+} \mathrm{e}^{\mathbf{A}^T t} \, [\mathbf{C}_\Omega]_i^T [\mathbf{C}_\Omega]_i \, \mathrm{e}^{\mathbf{A} t} \tag{5.98}$$

$$= \sum_{k=0}^{\infty} \left(\mathrm{e}^{\mathbf{A}^T \delta}\right)^k \mathbf{C}_\Omega^T \mathbf{C}_\Omega \left(\mathrm{e}^{\mathbf{A}\delta}\right)^k = \mathbf{Q}. \tag{5.99}$$

The last equality holds for the following reason. Since $\mathbf{A}$ is Hurwitz, $\mathrm{e}^{\mathbf{A}\delta}$ is Schur and $(\mathrm{e}^{\mathbf{A}\delta}, \mathbf{C}_\Omega)$ is observable according to (5.61). Inequality (5.99) is equivalent to

$$\mathbf{Q}^{-1} \preceq \mathbf{S}^{-1}. \tag{5.100}$$

Functions $\sqrt{\mathrm{Tr}(\mathbf{X})}$ and $\det(\log(\mathbf{X}))$ are nondecreasing on the cone of positive semi-definite matrices. By apply these functions to both sides of (5.100), one obtains the desired inequalities (5.62) and (5.63).

Chapter 6

Network Observability using Measurable Observations

6.1 Introduction

In Chapter 5, we consider *sampling* from certain locations across a continuous-time dynamical network. In this chapter, we look at the case where the noisy observations consist of measurable intervals of time. Such an assumption is a generalization of the continuous-time least-squares observers [38], in which the observations occur in a certain time-interval $[0, \tau]$ uniformly across the observation locations. To given an example, in Fig. 6.1, we show the classic observation strategy with $\tau = 0.12$ from 20 locations, while we are interested in observation times that have a typical arbitrary structure shown in the same figure.

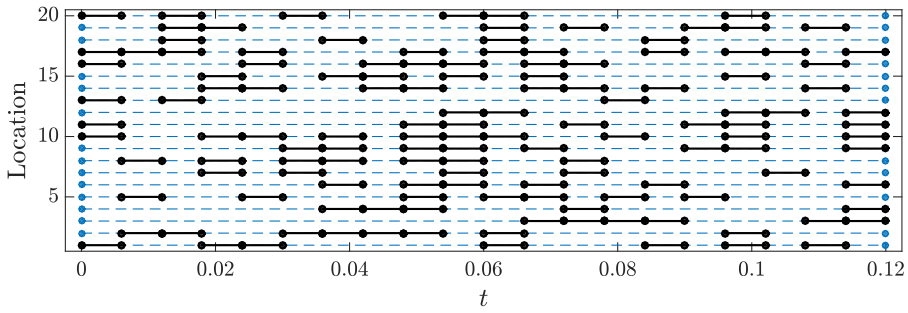

Figure 6.1: A sample illustration of the observation times in classic least-squares observer (dashed blue lines) and an arbitrary observation strategy (black solid lines).

6.2 Problem Statement

Consider a network that consists of n subsystems and each subsystems has the state $x_i \in \mathbb{R}$. The network state vector can be expressed by the vector

$$x := [x_1, \ldots, x_n]^T \in \mathbb{R}^n. \tag{6.1}$$

These subsystems are dynamically coupled through a linear time-invariant governing equation

$$\dot{x} = \mathbf{A}x, \tag{6.2}$$

with an initial condition $x_0 \in \mathbb{R}^n$. At time zero, we do not have any information on the initial state x_0. To estimate x_0, suppose that we can receive observations from a subset of the locations of the network $\Omega = \{i_1, \ldots, i_p\} \subset \{1, \ldots, n\}$, which we call observation locations. Then, we denote an observation strategy to be any set of the form

$$\mathfrak{S} = \left\{(i,t) \,\middle|\, \text{for every location } i \in \Omega : \ t \in \Theta_i\right\}, \tag{6.3}$$

where $\Theta_i \subset \mathbb{R}$ is the set of observation times at location $i \in \Omega$. Unlike Chapter 5, the set of observation times consists of something more than isolated points (more precisely, measurable subsets of time as we shortly assert). The corresponding family of observations is then

$$y := \big(x_i(t) + \xi_i(t) |\ \ (i,t) \in \mathfrak{S}\big), \tag{6.4}$$

where $\xi(t)$ is the white noise with variance σ^2. We can uniquely correspond Ω to an output matrix

$$C_\Omega := \left[e_{i_1} \mid \ldots \mid e_{i_p}\right]^T. \tag{6.5}$$

Again, we suppose that an estimation $\hat{x}_0$ to the initial state of the network x_0 is given and denote the error of the estimation by

$$\eta := \hat{x}_0 - x_0, \tag{6.6}$$

and the define the standard deviation of the error as an estimation measure, which is given by

$$\rho_d(\mathfrak{S}) := \mathbb{E}\left\{\sqrt{\eta^T \eta}\right\}. \tag{6.7}$$

The *research problem* of this chapter is to characterize properties of observation strategies[1] that allow us to recover initial state of linear network using sparse intervals of observation in space and time.

6.3 Characterizing Observation Strategies

We bring the basic definitions and results of the continuous frame theory. Then, we characterize the construction of the initial state of a linear network in this language.

6.3.1 Continuous Frame Theory

The contents of this subsection are mainly adapted from [151][2]. Suppose that a measure space $\mathfrak{S}$ is endowed by a nonnegative measure μ supported on $\mathfrak{S}$. Using this measure space, we define the basic operators that we will work with.

Definition 6.3.1. *Given a family of vectors $\Phi := (\phi_s)_{s\in\mathfrak{S}}$, the corresponding analysis operator $\mathbf{T}: \mathbb{R}^n \to L^2(\mathfrak{S}, \mu)$ is*

$$\mathbf{T}(x)(s) := \langle x, \phi_s \rangle, s \in \mathfrak{S},$$

[1] Compared to the observation strategies in the previous chapter.

[2] Almost all of the notions introduced in continuous frame theory have their own analogous concept in the discrete frame theory; for instance see the introductory materials given in [138]

and the frame operator $\mathbf{S}: \mathbb{R}^n \to \mathbb{R}^n$ *is*

$$\mathbf{S}(x) := \int_{\mathfrak{S}} \langle x, \phi_s \rangle \, \phi_s \, d\mu(s).$$

The analysis operator $\mathbf{T}$ has the adjoint operator $\mathbf{T}^*: L^2(\mathfrak{S}, \mu) \to \mathbb{R}^n$, which given by

$$\mathbf{T}^*(F) = \int_{\mathfrak{S}} F(s)\phi_s \, d\mu(s). \tag{6.8}$$

The analysis and frame operators are related to each other through the following identity:

$$\mathbf{S} = \mathbf{T}^*\mathbf{T}. \tag{6.9}$$

The frame operator $\mathbf{S}$ is a linear transformation from $\mathbb{R}^n$ to $\mathbb{R}^n$ and its canonical matrix representation is given by

$$\mathbf{S} = \int_{\mathfrak{S}} \phi_s \phi_s^T \, d\mu(s), \tag{6.10}$$

which we call the frame matrix. By (6.10), the frame matrix $\mathbf{S}$ is positive semi-definite, $\mathbf{S} = \mathbf{S}^T \succeq 0$. Next, we define the notion of continuous frame.

Definition 6.3.2. *A family of vectors* $\Phi = (\phi_s)_{s\in\mathfrak{S}}$ *in* $\mathbb{R}^n$ *is a frame for* $\mathbb{R}^n$ *if there exist two constants* $0 < \alpha \leq \beta$ *such that for each* $x \in \mathbb{R}^n$*, it holds that*

$$\alpha\|x\|_2^2 \leq \int_{\mathfrak{S}} |\langle x, \phi_s \rangle|^2 \, d\mu(s) \leq \beta\|x\|_2^2. \tag{6.11}$$

The largest lower frame bound and smallest upper frame bound are called the optimal frame bounds, which can be evaluated using the following result.

Proposition 6.3.3. *The family of vectors* $\Phi = (\phi_s)_{s\in\mathfrak{S}}$ *is a frame for* $\mathbb{R}^n$ *if and only if the frame matrix* $\mathbf{S}$ *is positive-definite. Moreover, the largest and smallest eigenvalues of frame matrix* $\mathbf{S}$ *associated with the frame* Φ *are the optimal frame bounds; i.e.,* $\alpha = \lambda_1(\mathbf{S})$ *and* $\beta = \lambda_n(\mathbf{S})$.

Consequently, the frame operator $\mathbf{S}$ is invertible if and only if the frame matrix in (6.10)

is invertible. It is of fundamental interest in the continuous frame theory to construct a vector $x \in \mathbb{R}^n$ using the family of observation y that is given by

$$y := y(s) = \mathbf{T}(x) \in L^2(\mathfrak{S}, \mu). \tag{6.12}$$

In the following proposition, we identify one way that we can reconstruct a given vector x from these observations.

Proposition 6.3.4. *If the family of vectors $\Phi = (\phi_s)_{s\in\mathfrak{S}}$ is a frame for $\mathbb{R}^n$, then vector $x \in \mathbb{R}^n$ inducing a family observations y given in (6.12) can be reconstructed via*

$$x = \mathbf{S}^{-1}\mathbf{T}^*(y(s)). \tag{6.13}$$

If we expand the reconstruction formula (6.13), we see that it has the matrix integral canonical representation,

$$x = \left(\int_{\mathfrak{S}} \phi_s \phi_s^T \, d\mu(s)\right)^{-1} \int_{\mathfrak{S}} y(s)\phi_s \, d\mu(s), \tag{6.14}$$

where family of observation vectors $y(s)$ is read from (6.12).

6.3.2 Exact Reconstruction of the Initial State

In what follows, we identify a frame that is suitable for construction of the initial state. We can write the solution to linear network (6.2) as $x(t) = \mathrm{e}^{\mathbf{A}t}x_0$ and hence we may write the components of the solution using the representation

$$x_i(t) = e_i^T \mathrm{e}^{\mathbf{A}t}x_0 = \left\langle x_0 \ , \ \mathrm{e}^{\mathbf{A}^T t}e_i \right\rangle \tag{6.15}$$

for each $i \in \Omega$. Because Ω is the set of observation locations and Θ_i is the set of observation times for each $i \in \Omega$, equation (6.15) motivates us to choose index set $\mathfrak{S}$ for the continuous frame to be

$$\mathfrak{S} = \{(i,t)|\ i \in \Omega,\ t \in \Theta_i\} \subset \Omega \times \mathbb{R}. \tag{6.16}$$

Moreover, for any observation label $(i,t) \in \mathfrak{S}$, it suggests to set the family of vectors associated to our frame as

$$\phi_s := \phi_{(i,t)} = \mathrm{e}^{\mathbf{A}^T t} e_i, \ s \in \mathfrak{S}. \tag{6.17}$$

For the upcoming developments, we consider the following assumption about the sets of observation times, which makes the integrals corresponding to the analysis and frame operators well-defined.

Assumption 6.3.5. *For each location $i \in \Omega$, the set of observation times Θ_i is a bounded Lebesgue-measurable set.*

Moreover, we consider index set of frame $\mathfrak{S}$ to be endowed by a nonnegative measure μ, which has the following structure (recall that μ_l denote the Lebesgue measure).

Assumption 6.3.6. *The nonnegative measure μ is a mixed measure on $\mathfrak{S}$ in the following sense: for each $i \in \Omega$, there exists a partial nonnegative measure μ_i that is the restriction of μ to $\{i\} \times \Theta_i \subset \mathfrak{S}$. Moreover, it holds that*

$$\mu_i\left(\{i\} \times \hat{\Theta}_i\right) = \mu_l\left(\hat{\Theta}_i\right) \quad \textit{for all l.m. subsets } \ \hat{\Theta}_i \subset \Theta_i;$$

i.e., μ_i acts similarly to the Lebesgue measure on the set $\hat{\Theta}_i \subset \Theta_i$. Moreover, $\mathrm{supp}(\mu_i) = \{i\} \times \Theta_i$.

The latter assumption also implies that

$$\mu(\{i\} \times \Theta_i) = \mu_i(\{i\} \times \Theta_i) = \mu_l(\Theta_i). \tag{6.18}$$

Under Assumptions 6.3.5 and 6.3.6, the matrix representation for the frame matrix $\mathbf{S}$ in (6.10) becomes

$$\mathbf{S} = \sum_{i \in \Omega} \int_{\Theta_i} \phi_{(i,t)} \phi_{(i,t)}^T \, d\mu_l(t). \tag{6.19}$$

The above setup for index set $\mathfrak{S}$ and its measure μ lets us characterize the conditions, under which, the construction of all initial states is feasible.

Theorem 6.3.7. *Suppose that Ω is the set of observation locations and $\Theta_i \subset \mathbb{R}$ is a set of observation times for each $i \in \Omega$ that satisfies Assumption 6.3.5. Consider an observation strategy $\mathfrak{S}$ given in* (6.16)*, which is endowed by a mixed measure μ that satisfies Assumption 6.3.6. Then, any initial state of the linear network* (6.2) *can be recovered based on the observation strategy $\mathfrak{S}$ if and only if the family of vectors*

$$\Phi = \left(\mathrm{e}^{\mathbf{A}^T t} e_i\right)_{(i,t)\in\mathfrak{S}} \tag{6.20}$$

endowed by the nonnegative measure μ is a frame of $\mathbb{R}^n$.

A set of locations Ω corresponds it to an output matrix,

$$\mathbf{C}_\Omega := \begin{bmatrix} e_{i_1} \mid \ldots \mid e_{i_p} \end{bmatrix}^T. \tag{6.21}$$

We next show that the observability is a necessary condition to be able to construct frames Φ of the form (6.20).

Theorem 6.3.8. *Consider an observation strategy $\mathfrak{S}$ that gives us a family of vectors Φ in* (6.20)*, which is a frame for $\mathbb{R}^n$. Then the pair $(\mathbf{A}, \mathbf{C}_\Omega)$ is observable.*

This results implies that if the observation locations result in an unobservable pair $(\mathbf{A}, \mathbf{C}_\Omega)$, then playing with the time labels of observations will not make it possible to construct every initial state. This justifies the following assumption.

Assumption 6.3.9. *The set of observation locations Ω is chosen such that the pair $(\mathbf{A}, \mathbf{C}_\Omega)$ is observable.*

The study of observability based on the choice of the locations in the network is a well-studied subject on its own. For instance, we refer to [152] and [121] for further reading.

Remark 19. The results of this Chapter does not depend on the network structure (e.g. sparsity pattern of $\mathbf{A}$). In fact, our results are for general linear time-invariant systems.

However, the use of observation locations are more meaningful in the context of the subsystems of a network, specifically when the subsystems are spatially distributed [153, 154].

6.4 Estimation under Noisy observations

Continuous observability frames with no observation noises are indistinguishable, because all of them provide us with exact construction of the initial state. However, in practice, the observations are always noisy and instead of exact reconstruction, we must consider the state estimation.

Recall that $\mathbf{T}$ is the analysis operator induced by the observability frame Φ, which is constructed using (i) sampling strategy $\mathfrak{S}$ as index set and (ii) nonnegative measure μ that has the desired properties discussed in the previous subsection. Suppose that instead of (6.12), the estimation is planned under a family of noisy observation vectors

$$y = \mathbf{T}(x_0) + \xi, \tag{6.22}$$

where ξ is a white noise with variance σ^2 at each location, while the noise processes are independent across those locations. Let us call the estimation to x_0 by $\hat{x}_0$. Then

$$\eta := \hat{x}_0 - x_0. \tag{6.23}$$

is the estimation error. The following result identifies the estimator that has the least-squares error (e.g. see [139]).

Theorem 6.4.1. *Suppose that we plan to estimate x_0, using the family of observations y given in (6.22). The estimator given by the formula*

$$\hat{x}_0 = \mathbf{S}^{-1}\mathbf{T}^*(y(s)) \tag{6.24}$$

is an unbiased estimator; i.e, $\mathbb{E}\{\hat{x}_0\} = x_0$. Furthermore, as an estimation measure, the

standard deviation of the error $\rho(\Phi) := \sqrt{\mathbb{E}\{\|\eta\|^2\}}$ *can be evaluated from the expression*

$$\rho(\Phi) = \sigma \left(\sum_{i=1}^{n} \lambda_i(\mathbf{S})^{-1} \right)^{1/2}. \tag{6.25}$$

We observe that this estimation measure is monotone, in the sense that if two observation strategies $\mathfrak{S}_1$ and $\mathfrak{S}_2$ satisfy $\mathfrak{S}_1 \subseteq \mathfrak{S}_2$, then the estimation qualities for the associated observability frames satisfy $\rho(\Phi_1) \geq \rho(\Phi_2)$.

Given an observation strategy, we can evaluate the total observation time across the network locations, which is

$$\mathfrak{t}(\mathfrak{S}) := \sum_{i \in \Omega} \mu_l(\Theta_i). \tag{6.26}$$

The quantity $\mathfrak{t}$ reflects the computational burden associated with the estimation problem, which bounds the estimation quality as shown below.

Theorem 6.4.2. *Given an observation strategy* $\mathfrak{S}$ *inducing an observability frame* Φ*, the estimation measure* $\rho(\Phi)$ *is bounded from below according to*

$$\rho(\Phi) \geq \frac{\sigma n}{\nu(\mathbf{A}, \mathfrak{S}) \sqrt{\mathfrak{t}(\mathfrak{S})}}, \tag{6.27}$$

where the total observation time $\mathfrak{t}$ *is given by* (6.26) *and*

$$\nu(\mathbf{A}, \mathfrak{S}) := \max_{(i,t) \in \mathfrak{S}} \left\| \mathrm{e}^{\mathbf{A}t} \right\|_2.$$

This theorem highlights a fundamental limit on the estimation measure, in the sense that given the maximum available observation time from all locations, $\mathfrak{t}(\mathfrak{S})$, the estimation measure can not be improved more than the illustrated limit no matter how we design the observation strategy.

6.5 Building Continuous Observation Strategies

While Theorem 6.3.7 brings the necessary and sufficient conditions for the reconstruction (or estimation), it does not provide us with actual construction procedures for these strategies and the corresponding frames. First, we bring our first result about construction of these frames.

Theorem 6.5.1. *Let the observation locations from the network be Ω and the set of observation times for each $i \in \Omega$ to be Θ_i. Consider the sampling strategy $\mathfrak{S}$ defined in* (6.16)*. Suppose that the following statements hold:*

- *the observation times Θ_i satisfy Assumption 6.3.5;*
- *the mixed nonnegative measure μ satisfies Assumption 6.3.6;*
- *the observation locations Ω satisfies Assumption 6.3.9.*

Then, the family of vectors Φ in (6.20) *defined over the set $\mathfrak{S}$ endowed by the nonnegative measure μ is a frame for $\mathbb{R}^n$.*

Remark 20. We observe that the assumption on the structure of the output matrix $\mathbf{C}_\Omega$ (where each output component is one of the states) is not necessary. In fact, the observation variables needs to correspond an observable output matrix $\mathbf{C}_\Omega$, whether they directly originate from the network locations or not (see Remark 19 as well).

We can use Theorem 6.5.1 to find simple designs for these continuous observability frames.

(i) Classical Least-Squares Estimator. Suppose that the observation horizon is $\tau > 0$ and the common observation interval is $\Theta_i = [0, \tau]$ for each $i \in \Omega$. Then, for $t \in [0, \tau]$, denote $y(t) = \mathbf{C}_\Omega x(t) + \xi(t)$. We can show that the corresponding least-squares estimator is given by

$$\hat{x}_0 = \left(\int_0^\tau \mathrm{e}^{\mathbf{A}^T t} \mathbf{C}_\Omega^T \mathbf{C}_\Omega \mathrm{e}^{\mathbf{A} t} \, dt \right)^{-1} \int_0^\tau \mathrm{e}^{\mathbf{A}^T t} \mathbf{C}_\Omega^T y(t) \, dt.$$

We have brought this matrix form, as it belongs to the classical least-squares initial state estimator (e.g. see [38]) for a system $\dot{x} = \mathbf{A}x$, where the noisy output y is observed. Thus, the classical estimator is a specific example of estimators that are under the study in this chapter.

(ii) Switching Observation Strategies. Let us consider the observation times for each location $i \in \Omega$ to be

$$\Theta_i = \bigcup_{j=1}^{M_i} \Theta_{ij} := \bigcup_{j=1}^{M_i} [t_{ij}, t_{ij} + \delta_{ij}], \tag{6.28}$$

where $M_i \geq 1$ and the positive numbers t_{ij} and δ_{ij} are chosen such that $t_{ij} + \delta_{ij} \leq t_{i(j+1)}$[3]. We call these the *switching observation strategies* and the corresponding frames the switching observability frames, because at each location, the observation becomes active only during certain time intervals. According to (6.19), the evaluation of the frame matrix in this case reduces to a double-sum

$$\mathbf{S} = \sum_{i \in \Omega} \sum_{j=1}^{M_i} \int_{\Theta_{ij}} \mathrm{e}^{\mathbf{A}^T t} e_i e_i^T \mathrm{e}^{\mathbf{A}t} \, dt \succ 0. \tag{6.29}$$

Remark 21. We observe that the classic estimator *(i)* can be considered as a special case of the switching observability frame *(ii)*, in infinitely many ways; for instance, suppose that in (6.28), for each $i \in \Omega$, we choose $M_i = M_0$, and for each $i \in \Omega$ and $j = 1, \ldots, M_0$, we select the non-overlapping observation intervals to be

$$\Theta_{ij} = \left[\frac{(j-1)\tau}{M_0}, \frac{j\tau}{M_0} \right]. \tag{6.30}$$

This discretization of the classic estimator is useful for the design of estimators whose observation intervals are only a subset of these intervals; i.e., the design of sparsified observability frames (see the next section).

We conclude this section by a simple example.

Example 6.5.2. Consider a two-dimensional system

$$\dot{x} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} x.$$

We choose the first state as the observation location; i.e., $\Omega = \{1\}$, $\mathbf{C}_\Omega = \begin{bmatrix} 1 & 0 \end{bmatrix}$. Thus

[3] Thus the observation intervals overlap at most at their limits.

$(\mathbf{A}, \mathbf{C}_\Omega)$ is observable. According to (6.28), we choose the interval $\Theta_1 = [t_1, t_2]$, for some $t_1 < t_2$. Then, using the formula (6.29), we compute the frame matrix $\mathbf{S}$, corresponding to these choices of observation location and time-interval. We can show that it has the determinant

$$\det(\mathbf{S}) = (t_1 - t_2)^2/4 - \sin(t_1 - t_2)^2/4.$$

We inspect that for $t_1 < t_2$, $\det(\mathbf{S}) > 0$, and therefore, $\mathbf{S} \succ 0$; i.e., the observation at the first location in the interval Θ_1 creates a frame for $\mathbb{R}^n$. Moreover, we can also show that

$$\rho(\Phi) = \sqrt{\sigma^2(t_2 - t_1)/\det(\mathbf{S})}.$$

6.6 Sparsification of Switching Strategies

Consider a given switching observation strategy inducing a switching observability frame with the time intervals given in (6.28). If the total time of the observations is higher than our computational/operational capacities, to enhance the scalability of the estimation using this frame, we would be interested in decreasing the duration of the observations in the observability frame. To do so, we define the sparsification of a switching observation strategy below.

Consider a switching observation strategy $\mathfrak{S}$ and its observability frame Φ with observation intervals that have the form of (6.28). Find a switching observation strategy $\mathfrak{S}_\mathrm{s}$ corresponding to a family of vectors Φ_s such that: *(i)* The observation times corresponding to $\mathfrak{S}_\mathrm{s}$ are

$$\Theta_i(\mathfrak{S}_\mathrm{s}) = \bigcup_{k=1}^{N_i} \Theta_{ij_k},$$

where $\{j_1, \ldots, j_{N_i}\} \subset \{1, \ldots, M_i\}$; i.e., a subset of the time-intervals of $\mathfrak{S}$ belong to $\mathfrak{S}_\mathrm{s}$. *(ii)* The number of observation time-intervals in $\mathfrak{S}_\mathrm{s}$ is less than $\mathfrak{S}$; i.e.,

$$\sum_{i \in \Omega} N_i < \sum_{i \in \Omega} M_i.$$

(iii) Φ_{s} is a frame for $\mathbb{R}^n$; i.e., it holds that

$$\mathbf{S}_{\mathrm{s}} = \sum_{i\in\Omega}\sum_{k=1}^{N_i}\int_{\Theta_{ij_k}} \mathrm{e}^{\mathbf{A}^T t} e_i e_i^T \mathrm{e}^{\mathbf{A}t}\, dt \succ 0.$$

Therefore, this notion of sparsification aims at trimming the observation times and consequently the support of frame measure μ, upon which, the sparsified frame Φ_{s} is evaluated. To fulfill these goals, we adapt the method of sparsification by effective resistances [51] and modify it to be able to find sparsifications to the switchings continuous frames. First, note that matrix $\mathbf{S}$ corresponding to a switching observability frame has the following number of switching components.

$$M := \sum_{i\in\Omega} M_i. \tag{6.31}$$

Adapting an arbitrary numbering for these components, we can write (6.29) equivalently as

$$\mathbf{S} = \sum_{k=1}^{M} \mathbf{\Psi}_k, \tag{6.32}$$

where the matrices $\mathbf{\Psi}_k \in \mathbb{R}^{n\times n}$ denote the arbitrary numbered integrals inside the summations of (6.29). For any component $\mathbf{\Psi}_k$, we define the leverage score by

$$r_k := \mathrm{Tr}\left(\mathbf{\Psi}_k \mathbf{S}^{-1}\right). \tag{6.33}$$

The sampling probability for this component is defined as

$$\pi_k := r_k/n. \tag{6.34}$$

We can see that these are indeed probabilities, because

$$\sum_{k=1}^{M}\pi_k = \frac{1}{n}\sum_{k=1}^{M}\mathrm{Tr}(\mathbf{\Psi}_k\mathbf{S}^{-1}) = \frac{1}{n}\mathrm{Tr}\left(\sum_{k=1}^{M}\mathbf{\Psi}_k\mathbf{S}^{-1}\right) = 1.$$

Now, for certain number of iterations, we sample the components $\mathbf{\Psi}_k$ with replacement,

Algorithm 6 Randomized Strategy Sparsification

Input: number of samples q, observation strategy $\mathfrak{S}$, observability frame Φ, switching components $\{\mathbf{\Psi}_j\}_{j=1,\dots,M}$

Output: observation strategy $\mathfrak{S}_\mathrm{s}$, switching components $\{\mathbf{\Psi}_{j_k}\}_{k=1,\dots,N}$, frame matrix $\mathbf{S}_\mathrm{s}$,

Initialize: $\mathbf{S}_\mathrm{s} = \mathbf{0}$, $\mathfrak{S}_\mathrm{s} = \varnothing$

  for $i = 1$ to q **do**

    randomly sample $\mathbf{\Psi}_k$ with probabilities $\pi_k = r_k/n$

    if $\mathbf{\Psi}_k$ has not been sampled before **then**,

      add corresponding location and times to $\mathfrak{S}_\mathrm{s}$

      update the frame $\mathbf{S}_s \leftarrow \mathbf{S}_\mathrm{s} + \mathbf{\Psi}_k$

    end if

  end for

  return Φ_s associated with $\mathfrak{S}_\mathrm{s}$, $\mathbf{S}_\mathrm{s}$

where the sampling probabilities are π_k given by (6.34). Then, if the component is not sampled before, we add it to the new frame. These steps give us Algorithm 6 with the following performance guarantee.

Theorem 6.6.1. *Suppose that we sparsify a switching observation strategy $\mathfrak{S}$ associated with the observability frame Φ according to Algorithm 6 with $q = O(n \log n/\epsilon^2)$. The output family of vectors Φ_s with probability more than one-half is a frame for $\mathbb{R}^n$. Moreover, with probability at least $1/4$, the estimation quality of using sparsified frame Φ_s is bounded according to*

$$\frac{\rho(\Phi_s) - \rho(\Phi)}{\rho(\Phi)} \leq 2\sqrt{\frac{\bar{\chi}}{1-\epsilon}} - 1, \tag{6.35}$$

with $\bar{\chi} := \mathbb{E}\{\chi\}$ where χ is a bounded random variable.

To see the proof and the definition of random variable χ, consult [155]. A earlier version of this theorem for rank-one sampled matrices is given in [156].

6.7 Numerical Examples

Example 6.7.1. We study a class of networks, wherein the magnitude of interaction between the subsystems decays exponentially by their distance. The state matrix of the network, $\mathbf{A} = [A_{ij}]$, is constructed as follows. We randomly choose a position for n subsystems in region $[0, 1]^2$. Then, we denote the distance between the positions of the subsystems i and

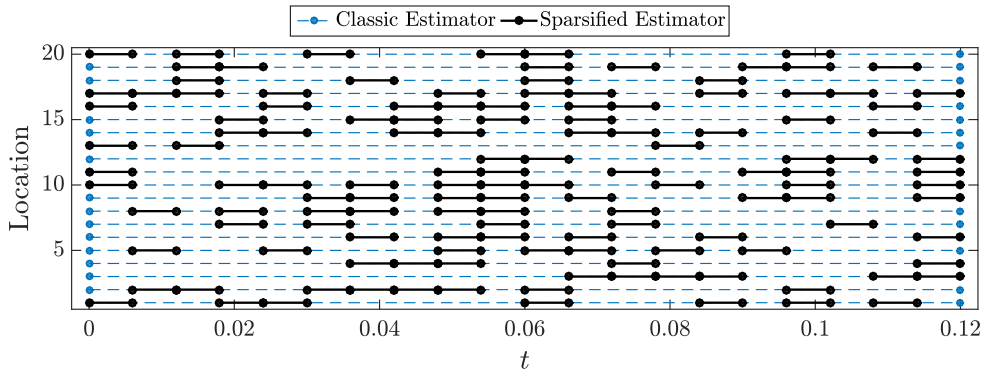

Figure 6.2: Space-time representation of the switching observation strategy (black lines) created by the sparsification of the observation strategy corresponding to the classical estimator (dashed blue lines) (see Example 6.7.1).

j by $\mathrm{dis}(i,j)$. For any $i,j \in \{1,\dots,n\}$, we set

$$A_{ij} = \begin{cases} c_{ij} \mathrm{e}^{-a\ \mathrm{dis}(i,j)^b} & \mathrm{dis}(i,j) \leq d/2 \\ 0 & \mathrm{dis}(i,j) > d/2 \end{cases}, \tag{6.36}$$

where $c_{ij} \sim \mathcal{N}(0,1)$ (independent for each element), d is the diameter of the connectivity disk around each subsystem, and parameters a and b control the decay properties of the information structure. We choose $n = 20$, $a = 1$, $b = 0.5$, and $d = 0.6$. Then, we make a random sample of this network and fix $\mathbf{A}$ for the developments of this example.

Classical Estimator: We set $\tau = 0.12$ and find the frame corresponding to the classic estimator, namely Φ using the whole state $\Omega = \{1,\dots,n\}$. For noise intensity $\sigma = 0.1$, we find that the estimation quality is $\rho(\Phi) \approx 1.2863$. The total observation time $\mathfrak{t}$ as given in (6.26) is $\mathfrak{t} = n\tau = 2.4$ units.

Sparsified Switching Estimator: As stated in Remark 21, the observation strategy corresponding to the classic estimator can be discretized, for instance, using (6.30). We set $M_i = M_0 = 20$ for each time-interval, and set Θ_{ij} based on (6.30). We get a switching observability frame with the of switching components, as defined by (6.31).

$$M = \sum_{i=1}^{n} M_0 = nM_0 = 400.$$

Next, we sparsify this switching observation frame using Algorithm 6 for $\epsilon = 0.6$ leading

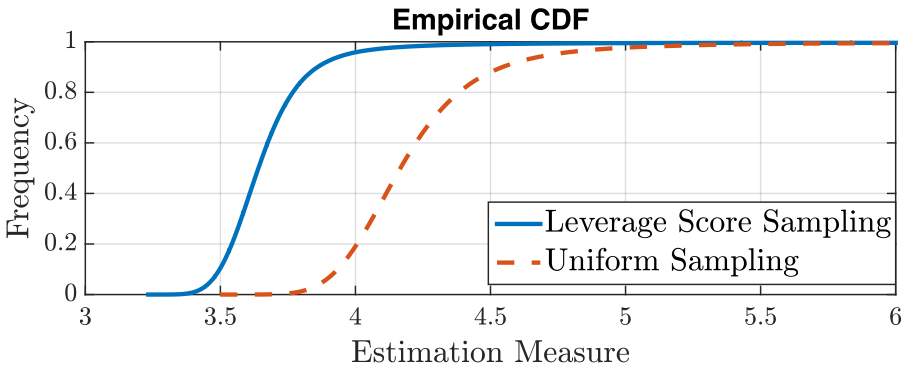

Figure 6.3: Empirical cdf for the estimation measure after sparsification using Algorithm 1 or completely at random (see Example 6.7.2).

to $q = 166$ samples. We observed $q_u = 134$ unique samples. The observation locations and times corresponding to the sparsified observation strategy $\mathfrak{S}_s$ are illustrated in Fig. 6.2. In this case $\rho(\Phi_s) \approx 2.3222$. Moreover, the total observation time $\mathfrak{t}$, in this case is

$$\mathfrak{t} = q_u \tau / M_0 \approx 0.8040.$$

Hence, as the result of sparsification, the total observation time is cut by almost two-thirds, while the estimation measure is less than double of the value for the classic estimator.

Example 6.7.2. We consider a consensus network of single-integrators agents that are connected over an undirected graph with Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$. We choose the graph to be an Erdős-Rényi random graph with 20 nodes and an edge probability of $p = 0.6$. We create a random graph with 109 edges and fix it for further developments. The dynamics of the network in this case are given by

$$\dot{x} = -\mathbf{L}x,$$

(e.g. see [157]). Again, we consider the discretized version of the classic estimator with $\tau = 0.1$. Then, we sparsify the resulting frames in two different ways: we use the original Algorithm 6 and a variation of Algorithm 6, in which instead of computing the probabilities based on the leverage scores, we uniformly choose the probabilities for all switching components. We run the sparsification with $\epsilon = 0.6$ for 5×10^5 different experiments. In Fig. 6.3,

we compare the experimental cumulative distribution function of the estimation measure, where the performance of the result of sparsification using the leverage scores is significantly superior to the uniform sampling (i.e., when we neglect the sampling probabilities proportional to the leverage scores).

6.8 Conclusion and Discussion

We have looked at the problem of measurable noisy observation from an observable subset of locations in the network for the initial state estimation. It turns out that the continuous frame formulation provides us with generalizations for the classic continuous least-square estimators.

Given a finite horizon and fixed subset of observation locations, the classic estimator over-performs the estimators built over observation strategies that only use the data at certain locations and certain periods. However, this results in a tradeoff between the quality of estimation and computational costs. One can use (6.27) to infer that this tradeoff should essentially scale with $\mathfrak{t}^{-1/2}$.

Appendix: Proofs

Proof of Theorem 6.3.8. Let us define

$$\tau_{\max} := \max_{t} \bigcup_{i\in\Omega} \Theta_i \text{ and } \tau_{\min} := \min_{t} \bigcup_{i\in\Omega} \Theta_i.$$

These two quantities are well-defined because of Assumption 6.3.5. Then, it holds that

$$\operatorname{rank}(\mathbf{S}) \leq \operatorname{rank}(\mathbf{S}^*), \tag{6.37}$$

where $\mathbf{S}^*$ is the matrix given by

$$\mathbf{S}^* := \int_{\tau_{\min}}^{\tau_{\max}} \mathrm{e}^{\mathbf{A}^T t} \mathbf{C}_\Omega^T \mathbf{C}_\Omega \mathrm{e}^{\mathbf{A} t}\, dt. \tag{6.38}$$

The reason is that $\mathbf{S}^*$ corresponds to a sampling strategy

$$\mathfrak{S}^* := \big\{(i,t)|\ i \in \Omega, t \in [\tau_{\min}, \tau_{\max}]\big\}, \tag{6.39}$$

where $\mathfrak{S} \subset \mathfrak{S}^*$. Thus, $\mathbf{S}^* \succeq \mathbf{S}$, which can be translated to the rank inequality (6.37). The family of vectors Φ is a frame by the assumption, which implies that

$$\mathbf{S} \succ 0 \text{ and } \tau_{\max} > \tau_{\min}. \tag{6.40}$$

Next, we write $\mathbf{S}^*$ as

$$\mathbf{S}^* = \mathrm{e}^{\mathbf{A}^T \tau_{\min}} \int_0^{\tau_{\max}-\tau_{\min}} \mathrm{e}^{\mathbf{A}^T t} \mathbf{C}_\Omega^T \mathbf{C}_\Omega \mathrm{e}^{\mathbf{A} t}\ dt\ \mathrm{e}^{\mathbf{A}\tau_{\min}}. \tag{6.41}$$

If the pair $(\mathbf{A}, \mathbf{C}_\Omega)$ is not observable, then

$$\operatorname{rank}(\mathbf{S}^*) < n, \tag{6.42}$$

because it is well-known that for a unobservable $(\mathbf{A}, \mathbf{C}_\Omega)$, rank of the matrix integral appearing in (6.41) is less than n. However, $\mathbf{S}$ is of full rank by (6.40). Therefore, (6.37) invokes a contradiction, implying that $(\mathbf{A}, \mathbf{C}_\Omega)$ is observable. □

Proof of Theorem 6.3.7. First we prove the sufficiency. Suppose that the family of vectors Φ endowed by the measure μ is a frame for $\mathbb{R}^n$. This implies that we can use the reconstruction equation (6.13) to find the initial state of the system.

Now, we prove the necessity. Suppose that the observation strategy induces a family of vectors that is not a frame for $\mathbb{R}^n$. This implies that the frame matrix $\mathbf{S}$ has rank deficiency. Therefore, for the initial states x_1 and $x_1 + x_2$ with a nonzero $x_2 \in \operatorname{null}(\mathbf{S})$, the resulting family of observations $y(s)$ given in (6.12) would be the same. Hence, the initial state reconstruction for $x_1 + x_2$ is infeasible. □

Proof of Theorem 6.4.2. The function $f(\lambda) = 1/\lambda$ is convex for positive numbers as its

second derivative is positive. Then, Jensen's inequality [158] implies that

$$\begin{aligned}\frac{1}{\sigma^2 n}\rho^2(\Phi) &= \frac{1}{n}\sum_{i=1}^{n} f(\lambda_i(\mathbf{S})) \\ &\geq f\left(\sum_{i=1}^{n}\frac{\lambda_i(\mathbf{S})}{n}\right) = f\left(\frac{\mathrm{Tr}(\mathbf{S})}{n}\right). \end{aligned} \tag{6.43}$$

We may bound the trace according to

$$\begin{aligned}\mathrm{Tr}\left(\mathbf{S}\right) &= \mathrm{Tr}\left(\sum_{i\in\Omega}\int_{\Theta_i}\phi_{(i,t)}\phi_{(i,t)}^T\, d\mu_l(t)\right) \\ &= \sum_{i\in\Omega}\int_{\Theta_i}\mathrm{Tr}\left(\phi_{(i,t)}\phi_{(i,t)}^T\right)\, d\mu_l(t) \\ &\leq \nu^2\sum_{i\in\Omega}\int_{\Theta_i} d\mu_l(t) = \nu^2\mathfrak{t}, \end{aligned} \tag{6.44}$$

where we used the fact that each t and $i \in \Omega$ we have

$$\begin{aligned}&\sqrt{\mathrm{Tr}\left(\phi_{(i,t)}\phi_{(i,t)}^T\right)} = \sqrt{\mathrm{Tr}\left(\phi_{(i,t)}^T\phi_{(i,t)}\right)} \\ &= \left\|\phi_{(i,t)}\right\|_2 = \left\|\mathrm{e}^{\mathbf{A}^T t}e_i\right\|_2 \leq \left\|\mathrm{e}^{\mathbf{A}^T t}\right\|_2\|e_i\|_2 \leq \nu.\end{aligned}$$

Combining (6.43) and (6.44), we get that

$$\rho^2(\Phi)/(\sigma^2 n) \geq f\left(\mathrm{Tr}(\mathbf{S})/n\right) \geq f\left(\nu^2\mathfrak{t}/n\right).$$

This concludes the proof, because $f(\lambda) = 1/\lambda$. □

Proof of Theorem 6.5.1. Recall that Assumption 6.3.5 together with Assumption 6.3.6 let us deduce the following property.

$$\mu(\{i\}\times\Theta_i) = \mu_i(\{i\}\times\Theta_i) = \mu_l(\Theta_i) > 0.$$

Hence, for each $i \in \Omega$, we partition every Θ_i according to

$$\Theta_i = \bigcup_{j=1}^{n}\Theta_{ij},$$

where Θ_{ij} are measurable sets with $\mu_l(\Theta_{ij}) > 0$. Moreover, for each $k, j = 1, \ldots, n$ with $k \neq j$, it holds that

$$\mu_l \left(\Theta_{ij} \cap \Theta_{ik}\right) = 0.$$

Based on this new partition, we observe that μ can be alternatively a refined mixed measure in this sense: there exists partial nonnegative measures μ_{ij}, where μ_{ij} is the restriction of μ to $\{i\} \times \Theta_{ij}$. Moreover, μ_{ij} acts similar to the Lebesgue measure on the set $\Theta_{ij} \subset \mathbb{R}$ and

$$\operatorname{supp}(\mu_{ij}) = \{i\} \times \Theta_{ij}.$$

Consider the vector $\theta = [t_{ij}]_{\substack{i\in\Omega \\ j=1,\ldots,n}}$ in $\mathbb{R}^{n|\Omega|}$. Now, we define the map of p.s.d. matrices $\mathbf{S}_d(\theta)$ from $\mathbb{R}^{n|\Omega|}$ to be

$$\mathbf{S}_d(\theta) := \sum_{i\in\Omega} \sum_{j=1}^{n} \phi_{(i,t_{ij})}.$$

In [156], we have shown that if Assumption 6.3.9 holds, then the Lebesgue measure of the points in $\mathbb{R}^{n|\Omega|}$ for which $\mathbf{S}_d(\theta)$ does not have full rank is zero. On the other hand, note that the frame matrix can be written as

$$\mathbf{S} = \sum_{i\in\Omega} \sum_{j=1}^{n} \int_{\Theta_{ij}} \phi_{(i,t_{ij})} \, d(t_{ij}).$$

Because of linearity of the integral and previous argument we can conclude that $\mathbf{S} \succ 0$, thus Φ is a frame for $\mathbb{R}^n$. $\square$

Chapter 7

Fast Landmark Selection in Robot Visual Navigation

7.1 Introduction

Safe and robust navigation in uncertain environments is one of the fundamental problems in robotics. The recent technological advances in the computing devices have opened up new opportunities and made several breakthroughs possible in this research area [159], where estimation and planning problems can be solved close to real-time in some applications. However, robot navigation in rapidly changing environments still suffer from computational complexities. Even if one uses high-performance computational units, the demand for agility and higher levels of autonomy always mandates us to execute onboard procedures in shorter periods of time.

One of the essential subproblems during robot navigation is to solve the localization, mapping, and visual odometry at an acceptable level of accuracy while spending a minimal amount of computational resources [160]. To achieve this goal, many researchers have investigated visual feature selection problem [161–168]. The underlying idea is that depending on the current state of the robot and planned motion in the near future (i.e., the task), tracking certain features across a time horizon can be more informative than tracking other features. In other words, certain visual features may deserve more attention compared to the rest. In

this regard, [169] uses a greedy method to select a subset of pre-identified visual landmarks which facilitate the pose estimation of the robot. In [165], the authors combine solving the simultaneous localization and mapping (SLAM) using the unscented Kalman filtering with reinforcement learning. Their approach generates policies that govern the feature selection. In [168], a two-stage methodology for measurement planning is discussed. The first stage is the selection of the subset of landmarks for observation, which is followed by the design of observation times for each feature. In [163], the authors consider the problem of task-aware design of a subset of features such that an uncertainty metric is minimized. In [170], the authors consider a visual-inertial navigation problem and analyze the problem of feature-selection, where the design variable is the features that will be tracked during a fixed time-horizon. They use convex relaxations as well as the greedy method for feature selection and quantify performance guarantees for the quality of the resulting estimations.

In this Chapter, we propose a method to reduce time complexity of the feature selection subproblem during the navigation. The navigation setup consists of a robot that moves based on generated position estimates. The robot is assumed to use an onboard camera to (passively) track selected features over a fixed time horizon to improve the quality of the estimation. Our utilized model for vision system is similar to that of [170], while instead of using the greedy method and convex relaxations, we propose a randomized sampling algorithm for feature selection. In our approach, a sampling probability (a number between 0 and 1) is assigned to each available feature, where our randomized algorithm interprets these numbers as a measure of informativeness during sampling process (a feature is more informative if its sampling probability is closer to 1). Several theoretical guarantees on the quality of the estimation is derived. It turns out that time-complexity of our randomized sampling algorithm scales linearly with the number of available features, while time complexity of the greedy method of [170] scales quadratically for the exact same problem. It should be emphasized that our algorithm is more suitable for problems with hundreds or thousands of features, where greedy methods become practically inefficient. Numerical simulations confirm that the estimation quality using features provided by our randomized sampling is very close to the quality of estimation provided by the greedy method, while the required time to run our randomized sampling algorithm is significantly less than the

greedy method.

After stating the problem in Section 7.2, we discuss the details of the motion and vision models in Section 7.3. Then, three estimation measures are introduced to quantify the quality of estimation based on the selected features. In Section 7.5, we propose a randomized algorithm for feature selection and conduct performance and time-complexity analysis for our approach. In Section 7.6, a numerical case is explained, wherein we compare the performance of our (weighted) randomized feature selection method against: (i) a randomized method that selected features uniformly, and (ii) the greedy method. The proofs of all theoretical results can be found in [155].

Notations: The set of nonnegative integer and real numbers are denoted by $\mathbb{Z}_+$ and $\mathbb{R}_+$, respectively. The vectors and matrices are denoted by lower-case and upper-case letters, respectively (e.g. x and X). The identity matrix of size n is denoted by I_n. The set of positive definite matrices of size n is denoted by $\mathcal{S}^n_{++}$. The partial ordering on the cone of positive-semidefinite matrices is denoted by $\succ$, $\succcurlyeq$, $\prec$, and $\preccurlyeq$ operators. The block-diagonal matrix with diagonal elements $X_1, \ldots, X_N$ is denoted by $\mathrm{diag}(X_1, \ldots, X_N)$. For a set S, $|S|$ denotes its cardinality. For a map $\mathbf{g}$, $\nabla_x \mathbf{g}$ denotes the corresponding partial derivative. A Gaussian random variable with mean vector μ and covariance matrix Σ is denoted by $\mathcal{N}(\mu, \Sigma)$. $X \otimes Y$ denotes the Kronecker product of matrices X and Y. The special orthogonal group in 3 dimensions is denoted by SO(3).

7.2 Feature Selection Problem

Let us denote spatial location of a robot at time $t \in \mathbb{Z}_+$ by $x_t \in \mathbb{R}^3$. For a given positive integer T, the vector of future states over the discrete time horizon $[t, t+T] = t, t+1, \ldots, t+T$ is represented by

$$\mathbf{x}_{t,T} := \left[x_t^T,\ x_{t+1}^T,\ \ldots\ ,\ x_{t+T}^T\right]^T \in \mathbb{R}^{3(T+1)}.$$

Since robot motion creates uncertainty, having access to the statistics of $\mathbf{x}_{t,T}$ will help us measure quality of our prediction of robot whereabouts over the time horizon [170]. As it

is shown in Subsection 7.3.1, one can obtain mean vector $\bar{\boldsymbol{\mu}}_{t,T} \in \mathbb{R}^{3(T+1)}$ and covariance matrix $\bar{\boldsymbol{\Sigma}}_{t,T} \in \mathcal{S}_{++}^{3(T+1)}$ of $\mathbf{x}_{t,T}$ under popular Gaussianity assumption. These quantities can be equivalently transformed into more relevant forms for the feature selection problem, namely, information vector and matrix, which are given by [160]

$$\bar{\mathbf{b}}_{t,T} = \bar{\boldsymbol{\mu}}_{t,T}^T \bar{\boldsymbol{\Sigma}}_{t,T}^{-1} \tag{7.1}$$

$$\bar{\mathbf{H}}_{t,T} = \bar{\boldsymbol{\Sigma}}_{t,T}^{-1}. \tag{7.2}$$

There is a one-to-one correspondence between mean vector and covariance matrix and their counterparts information vector and information matrix. A striking property of the latter representation is that the contribution of each feature (or landmark) to the information vector and matrix is fused linearly [171]. Having the prior estimation parameters (7.1)-(7.2), as it is shown in Subsection 7.3.2, the quality of estimation for $\mathbf{x}_{t,T}$ can be improved by fusing information of newly observed visual features using an onboard camera. The updated information matrix and vector are

$$\mathbf{H}_{t,T}(\Theta_t) = \bar{\mathbf{H}}_{t,T} + \sum_{f \in \Theta_t} \mathbf{H}_{t,T}^f \tag{7.3}$$

$$\mathbf{b}_{t,T}(\Theta_t) = \bar{\mathbf{b}}_{t,T} + \sum_{f \in \Theta_t} \mathbf{b}_{t,T}^f \tag{7.4}$$

in which $\mathbf{b}_{t,T}^f$ and $\mathbf{H}_{t,T}^f$ are contributions of feature f to the overall information matrices of the estimation problem. The set of all identifiable features (landmarks) at time t, which can be triangulated using multiple frames over the time horizon $[t, t+T]$, is denoted by Θ_t. Suppose that $|\Theta_t| = N_t$ is assumed to be large.

Tracking a large number of features (landmarks) for accurate navigation usually requires substantial onboard computational power [162]. As a result, a desirable navigation objective is to select and track a small subset of features that are more informative, while providing an acceptable estimation quality. Suppose that robot is only capable of tracking at most q, which is comparably less than N_t, features during the horizon.

Definition 7.2.1. *A map $\rho : \mathcal{S}_{++}^n \to \mathbb{R}$ is called monotone decreasing if $X \preceq Y$ implies*

$\rho(X) \geq \rho(Y)$.

Then, the feature selection problem can be formulated as

$$\underset{\Phi_t \subset \Theta_t}{\text{minimize}} \quad \rho\big(\mathbf{H}_{t,T}(\Phi_t)\big) \tag{7.5}$$

$$\text{subject to}: \quad |\Phi_t| \leq q \tag{7.6}$$

where $\rho: \mathcal{S}_{++}^{3(T+1)} \to \mathbb{R}$ is a monotone decreasing map that measures the estimation quality.

The optimization problem (7.5)-(7.6) is combinatorial and usually NP-hard. The *research problem* is to propose a scalable algorithm that provides solutions for (7.5)-(7.6) with performance guarantees.

7.3 Models for Robot Motion and Vision System

In order to calculate faithful estimates for the robot's position, one needs to properly fuse estimates resulting from models of robot motion with the estimates resulting from camera models.

7.3.1 Statistics of Robot Position

The goal is to calculate information vector and matrix of $\mathbf{x}_{t,T}$ when dynamics of robot evolves over time horizon $[t, t+T]$. To achieve this, we utilize a model that is inspired by the dynamic model analyzed in [171]. Suppose that dynamics of the robot's position is governed by

$$x_\tau = \mathbf{g}(x_{\tau-1}, u_\tau) + \delta_\tau \tag{7.7}$$

for all $\tau \in [t+1, t+T]$, where $\mathbf{g}: \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$ is a (possibly nonlinear) known map, where u_τ in the control command at time τ, and $\delta_\tau \sim \mathcal{N}(0, \Lambda_\tau)$ is a temporally independent random process that captures the aggregate effect of all uncertainties induced by the robot

motion. It is assumed that a feedback control law with the following structure is given

$$u_\tau = \mathbf{h}(u_\tau^{\text{ref}}, x_{\tau-1}) \tag{7.8}$$

that ensures the robot with dynamics (7.7) tracks a reference path, at least in the absence of uncertainties, with some desired accuracy. The command u_τ^{ref} may have to be pre-filtered to enhance the tracking performance.

Let us represent the actual robot's position by$x_{\tau-1}$, which is a random variable and its true value is unknown. We use its mean value $\bar{\mu}_{\tau-1}$ as a faithful estimate of its value in (7.8) to obtain

$$\bar{u}_\tau = \mathbf{h}\left(u_\tau^{\text{ref}}, \bar{\mu}_{\tau-1}\right). \tag{7.9}$$

In presence of uncertainties, the trajectory of the closed-loop system (7.7)-(7.9) will fluctuate around the reference path and the tracking quality will depend on the quality of estimation $\bar{\mu}_{\tau-1}$. The control mechanism (7.9) is merely using the initial statistics of position of the robot. In the next subsection, we show that incorporating new information obtained from observing features (landmarks) will help us improve the estimation quality, which in turn will improve the path tracking quality.

Suppose that the current, i.e., before accounting for the dynamics of the robot, pose estimates for x_t is described by mean vector μ_t and covariance matrix Σ_t. For the time step starting at $\tau = t$ in the horizon, let us set $\bar{\mu}_t = \mu_t$ and $\bar{\Sigma}_t = \Sigma_t$. For the next steps, we define the composed map

$$\mathbf{f}(x, \mu, u) := \mathbf{g}\big(x, \mathbf{h}(u, \mu)\big). \tag{7.10}$$

Then, upon linearizing the dynamics of the system at working point $(x, \mu, u) = (\bar{\mu}_{\tau-1}, \bar{\mu}_\tau, u_\tau^{\text{ref}})$ with respect to x, we get

$$x_\tau \;\approx\; \bar{\Delta}_\tau + A_\tau \left(x_{\tau-1} - \bar{\mu}_{\tau-1}\right) + \delta_\tau \tag{7.11}$$

in which vector $\bar{\Delta}_\tau$ and A_τ are given by

$$\bar{\Delta}_\tau := \mathbf{g}\big(\bar{\mu}_{\tau-1}, \mathbf{h}(u_\tau^{\text{ref}}, \bar{\mu}_{\tau-1})\big) \tag{7.12}$$

$$A_\tau := \nabla_x \mathbf{f}(\bar{\mu}_{\tau-1}, \bar{\mu}_{\tau-1}, u_\tau^{\text{ref}}) \tag{7.13}$$

for all $\tau \in [t+1, t+T]$.

Lemma 7.3.1. *By setting $\bar{\mu}_t = \mu_t$ and $\bar{\Sigma}_t = \Sigma_t$, the mean and covariance of $\mathbf{x}_{t,T}$ is given by*

$$\bar{\mathbf{\Sigma}}_{t,T} = \begin{bmatrix} \bar{\Sigma}_t & \bar{\Sigma}_{t,t+1} & \dots & \bar{\Sigma}_{t,t+T} \\ \bar{\Sigma}_{t,t+1}^T & \bar{\Sigma}_{t+1} & \dots & \bar{\Sigma}_{t+1,t+T} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\Sigma}_{t,t+T}^T & \bar{\Sigma}_{t+1,t+T}^T & \dots & \bar{\Sigma}_{t,T} \end{bmatrix} \tag{7.14}$$

$$\bar{\boldsymbol{\mu}}_{t,T} = \begin{bmatrix} \bar{\mu}_t^T & \bar{\mu}_{t+1}^T & \dots & \bar{\mu}_{t+T}^T \end{bmatrix}^T \tag{7.15}$$

where

$$\bar{\Sigma}_\tau = A_\tau \bar{\Sigma}_{\tau-1} A_\tau^T + \Lambda_\tau$$

$$\bar{\mu}_\tau = \bar{\Delta}_\tau$$

for every instant $\tau \in [t+1, t+T]$ and

$$\bar{\Sigma}_{\tau_1,\tau_2} = \left(\prod_{i=1}^{\tau_2-\tau_1} A_{\tau_2-i-1} \right) \bar{\Sigma}_{\tau_1}$$

for all $\tau_1, \tau_2 \in [t, t+T]$ with $\tau_1 < \tau_2$.

We can substitute (7.14) and (7.15) into (7.1) and (7.2) to calculate information vector $\bar{\mathbf{b}}_{t,T}$ and matrix $\bar{\mathbf{H}}_{t,T}$. In the next subsection, it is shown that these vectors and matrices will be updated upon receipt of certain information about the observed features over the time horizon $[t, t+T]$.

7.3.2 Camera Model for Feature Tracking and Estimation

We employ the observation model proposed by in [170] for an onboard camera. For every $\tau \in [t, t+T]$, let us denote orientation of the robot by rotation matrix $R_\tau \in \mathrm{SO}(3)$, orientation of the camera with respect to the robot by rotation matrix $R_{\mathrm{c}} \in \mathrm{SO}(3)$, translation of the camera with respect to the robot pose by $x_{\mathrm{c}} \in \mathbb{R}^3$, the unit vector corresponding to pixel measurement of feature $f \in \Theta_t$ at time τ by $u^f_{\tau,T} \in \mathbb{R}^3$, and the position vector of the feature by $y_f \in \mathbb{R}^3$. We recall that the corresponding skew-symmetric matrix induced by $u^f_{\tau,T}$ satisfies

$$U^f_{\tau,T}\, v = u^f_{\tau,T} \times v$$

for every vector $v \in \mathbb{R}^3$. As it is discussed in [170], one may reasonably assume that the observation vector in the image and its counterpart in real world are collinear (i.e., parallel). However, due to existence of noise in the process, one may consider a disrupted version of this assumption by considering the following noisy observation model

$$U^f_{\tau,T}\left(\left(R_\tau R_{\mathrm{c}}\right)^T \left(y_f - \left(x_\tau + R_\tau x_{\mathrm{c}}\right)\right)\right) = \eta^f_{\tau,T}, \tag{7.16}$$

where $\eta^f_{\tau,T} \sim \mathcal{N}(0, \sigma^2 I_3)$. The observation model (7.16) can be rewritten as

$$z^f_{\tau,T} = U^f_{\tau,T}\left(R_\tau R_{\mathrm{c}}\right)^T \left(x_\tau - y_f\right) + \eta^f_{\tau,T} \tag{7.17}$$

with $z_{f,\tau} := -U_{f,\tau} R_c^T x_c = U^T_{f,\tau} R_c^T x_c$. The camera takes one frame at every time instant over time horizon $[t, t+T]$. With knowledge of planned motion (i.e., location and orientation) for robot over the time horizon, suppose that robot is capable of running forward simulations to determine a feature will be visible in n_f frames out of all $T+1$ frames over the time horizon. By considering relation (7.17) for such visible features, one can stack all these equations and write them in more compact form

$$\mathbf{z}^f_{t,T} = \mathbf{F}^f_{t,T} \mathbf{x}_{t,T} + \mathbf{E}^f_{t,T} y_f + \boldsymbol{\eta}^f_{t,T}, \tag{7.18}$$

for some appropriate matrices $\mathbf{F}_{t,T}^{f}$ and $\mathbf{E}_{t,T}^{f}$. A given apriori information matrix $\bar{\mathbf{H}}_{t,T}$, which is obtained from (7.14), can be updated by fusing information of a visible feature $\{f\}$ according to the following rule [170]

$$\mathbf{H}_{t,T}(\{f\}) = \bar{\mathbf{H}}_{t,T} + \mathbf{H}_{t,T}^{f}, \tag{7.19}$$

where the linearly added information matrix is given by

$$\mathbf{H}_{t,T}^{f} = \sigma^{-2}\Big(\left(\mathbf{F}_{t,T}^{f}\right)^{T} \mathbf{F}_{t,T}^{f} - \left(\mathbf{F}_{t,T}^{f}\right)^{T} \mathbf{E}_{t,T}^{f} \left(\left(\mathbf{E}_{t,T}^{f}\right)^{T} \mathbf{E}_{t,T}^{f} \right)^{-1} \left(\mathbf{E}_{t,T}^{f}\right)^{T} \mathbf{F}_{t,T}^{f} \Big).$$

This additive property of the information matrix can be verified by application of the Bayes law [171] together with the Schur complement [170]. A similar treatment allows us to derive the following update rule for the information vector.

Lemma 7.3.2. *The information vector of $\mathbf{x}_{t,T}$ upon tracking feature $f \in \Theta_t$ is updated according to*

$$\mathbf{b}_{t,T}(\{f\}) = \bar{\mathbf{b}}_{t,T} + \left(\mathbf{B}_{t,T}^{f} \mathbf{z}_{t,T}^{f}\right)^{T}, \tag{7.20}$$

where matrix $\mathbf{B}_{t,T}^{f}$ is given by

$$\mathbf{B}_{t,T}^{f} := \sigma^{-2} \left(\left(\mathbf{F}_{t,T}^{f}\right)^{T} - \left(\mathbf{F}_{t,T}^{f}\right)^{T} \mathbf{E}_{t,T}^{f} \left(\left(\mathbf{E}_{t,T}^{f}\right)^{T} \mathbf{E}_{t,T}^{f} \right)^{-1} \left(\mathbf{E}_{t,T}^{f}\right)^{T} \right).$$

Since contributions of different features are independent of each other, for a selected subset of features $\Phi_t \subset \Theta_t$, one can verify that the updates to the information matrix and vector upon the choice of these features are given by

$$\mathbf{H}_{t,T}(\Phi_t) = \bar{\mathbf{H}}_{t,T} + \sum_{f \in \Phi_t} \mathbf{H}_{t,T}^{f}, \tag{7.21}$$

$$\mathbf{b}_{t,T}(\Phi_t) = \bar{\mathbf{b}}_{t,T} + \sum_{f \in \Phi_t} \left(\mathbf{B}_{t,T}^{f} \mathbf{z}_{t,T}^{f}\right)^{T}. \tag{7.22}$$

The corresponding mean vector and covariance matrix for the localization problem can be

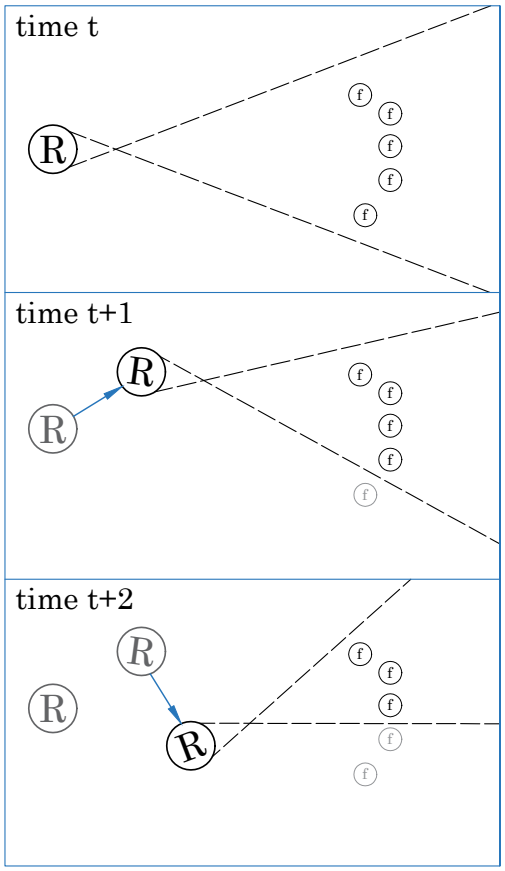

Figure 7.1: The schematic of the motion and vision model adopted in this chapter at three consecutive snapshots. The location of the robot is denoted by letter R, which moves and changes its orientation across three frames. The features are denoted by letter f. As a result of this movement, the visible features (those in between the dashed lines) in each frame will vary. Set Θ_t will consist of all features that can be triangulated during this horizon.

calculated through

$$\boldsymbol{\Sigma}_{t,T}(\Phi_t) = \mathbf{H}_{t,T}(\Phi_t)^{-1}, \tag{7.23}$$

$$\boldsymbol{\mu}_{t,T}(\Phi_t)^T = \mathbf{b}_{t,T}(\Phi_t)\,\mathbf{H}_{t,T}(\Phi_t)^{-1}. \tag{7.24}$$

In Fig. 7.1, we illustrate the essence of the motion and vision model described in this section. In order to specify the set of available features Θ_t for tracking, the robot needs to determine which features can be triangulated and which ones will result in invertible information matrices $(\mathbf{E}^f_{t,T})^T\mathbf{E}_f$ (cf. [170]).

7.4 Estimation Measures

Given a subset of trackable features Φ_t, we can quantify the quality of resulting estimation in various meaningful ways.

(i) Variance of the Error: Given the covariance matrix, the variance of error equals the sum of the variances of all scalar components of vector $\mathbf{x}_{t,T}$. This measure can be characterized as

$$\rho_v(\mathbf{H}_{t,T}(\Phi_t)) := \mathrm{Tr}(\mathbf{H}(\Phi_t)^{-1}) = \mathrm{Tr}(\boldsymbol{\Sigma}(\Phi_t)). \tag{7.25}$$

(ii) Differential Entropy of the Estimation Error: It is known that the differential entropy of a multivariate Gaussian random variable with covariance $\boldsymbol{\Sigma}$ is

$$h = \frac{1}{2}\log(\det(\boldsymbol{\Sigma})) + \frac{n}{2}\big(1 + \log(2\pi)\big).$$

This measure quantifies the uncertainty volume of the estimation error, which is given by

$$\rho_e(\mathbf{H}_{t,T}(\Phi_t)) = \log(\det(\boldsymbol{\Sigma}(\Phi_t))) = -\log(\det(\mathbf{H}_{t,T}(\Phi_t))),$$

(iii) Spectral Variance: Let us consider the eigen-space of the largest eigenvalue of the covariance matrix of the estimator. This is the subspace across which the estimation is less

accurate than the rest of the directions. Thus, we can use the following estimation measure

$$\rho_\lambda(\mathbf{H}_{t,T}(\Phi_t)) = \lambda_{\max}(\mathbf{\Sigma}(\Phi_t)) = \lambda_{\min}(\mathbf{H}_{t,T}(\Phi_t))^{-1}.$$

All these measures are monotonically decreasing. They are also spectral functions, i.e., they only depend on the eigenvalues of the information or covariance matrices. Therefore, having lower and upper bounds for the covariance matrix can be potentially useful to obtain similar bounds for these estimation measures. Measures *(ii)* and *(iii)* have been also discussed in [170].

7.5 Feature Selection via Randomized Sampling

We propose a scalable algorithm that provides feasible solutions for (7.5)-(7.6) with provable performance bounds.

7.5.1 Leverage Scores and Induced Probabilities

Each available feature $f \in \Theta_t$ is assigned some nonnegative numbers, which are so-called leverage scores, that are closely related to the notion of effective resistances in graph sparsification problem [51]. The maximal information matrix of $\mathbf{x}_{t,T}$, over the cone of positive-definite matrices, corresponds to the case where all features are employed in the estimation process. This matrix is given by

$$\mathbf{H}_{t,T}(\Theta_t) = \bar{\mathbf{H}}_{t,T} + \sum_{f \in \Theta_t} \mathbf{H}_{t,T}^f. \tag{7.26}$$

Therefore, for any $\Phi_t \subset \Theta_t$, it holds that

$$\mathbf{H}_{t,T}(\Phi_t) \preceq \mathbf{H}_{t,T}(\Theta_t). \tag{7.27}$$

Hence, for every monotone decreasing map $\rho : \mathcal{S}_{++}^{3(T+1)} \to \mathbb{R}$, it follows that

$$\rho(\mathbf{H}_{t,T}(\Theta_t)) \leq \rho(\mathbf{H}_{t,T}(\Phi_t)).$$

Using the maximal matrix, we define

$$\bar{\mathbf{H}}^f_{t,T} := \frac{1}{N_t}\bar{\mathbf{H}}_{t,T} + \mathbf{H}^f_{t,T}. \tag{7.28}$$

for every $f \in \Theta_t$ with $|\Theta_t| = N_t$.

Definition 7.5.1. *For a given set of features Θ_t, the leverage scores are nonnegative numbers that are defined by*

$$r_f := \mathrm{Tr}\left(\mathbf{H}_{t,T}(\Theta_t)^{-1}\bar{\mathbf{H}}^f_{t,T}\right) \tag{7.29}$$

for every feature $f \in \Theta_t$.

One can associate a probability mass function denoted by $\pi : \Theta_t \to [0,1]$ to elements of Θ_t by setting

$$\pi(f) = \pi_f = \frac{r_f}{n}, \tag{7.30}$$

where $n = 3(T+1)$ is the dimension of $\bar{\mathbf{H}}_{t,T}$. The resulting function is a well-defined probability mass function as we have

$$\sum_{f\in\Theta_t} \pi(f) = \frac{1}{n}\sum_{f\in\Theta_t} \mathrm{Tr}\left(\mathbf{H}_{t,T}(\Theta_t)^{-1}\bar{\mathbf{H}}^f_{t,T}\right) = 1.$$

7.5.2 Sampling Algorithm

The steps of our method are given in Algorithm 7. First, we iteratively and independently sample a feature from Θ_t with replacement for q iterations. This sampling takes place according to probability mass function π, which is defined by (7.30). The sampled feature f is added to Φ_t provided that it has not been sampled before. At the end of the procedure, set of selected features Φ_t will have at most q elements.

Remark 22. Our approach is inspired by graph sparsification methods using effective resistances [51, 57]. A special form of our method in this chapter appears in [156] for selecting rank-one matrices when the constant term $\bar{\mathbf{H}}_{t,T}$ is zero.

Remark 23. Our randomized feature selection algorithm does not depend on robot motion; see Section 7.3.1 for more details. For instance, our feature selection approach can be used

Algorithm 7 Randomized Feature Selection

input: initial information matrix $\bar{\mathbf{H}}_{t,T}$
set of available features Θ_t, number of samples q
output: selected features Φ_t, information matrix $\mathbf{H}_{t,T}$
initialize: $\Phi_t = \varnothing$, $\mathbf{H}_{t,T} = \bar{\mathbf{H}}_{t,T}$
for $k = 1$ to q **do**
sample a feature from Θ_t using distribution $\pi \to f$
select the corresponding matrix

$$\mathbf{H} \leftarrow \mathbf{H}_{t,T}^{f}$$

if $f \notin \Phi_t$, **then**
add f to Φ_t
update the information matrix:

$$\mathbf{H}_{t,T} \leftarrow \mathbf{H}_{t,T} + \mathbf{H}$$

end if
end for

instead of the feature selection routines in the inertial-visual navigation setup described in [170] that are based on the greedy method and convex relaxations.

7.5.3 Performance Guarantee

Algorithm 7 provides us with an information matrix that is a constant-factor approximation to the maximal information matrix $\mathbf{H}_{t,T}(\Theta_t)$ given by (7.26).

Theorem 7.5.2. *For a given parameter $\epsilon \in (0, 1)$, suppose that Algorithm 7 is executed with a fixed $q = O(n \log n/\epsilon^2) < N_t$. Then, the resulting information matrix, see* (7.21)*, based on the resulting set of features Φ_t, satisfies*

$$\mathbf{H}_{t,T}(\Phi_t) \succeq \frac{1-\epsilon}{4\bar{\chi}}\, \mathbf{H}_{t,T}(\Theta_t). \tag{7.31}$$

with probability at least $1/4$ for a number $\bar{\chi}$.

The proof of this theorem and definition of $\bar{\chi}$ is rather involved and inspired by [51]. The spectral bound (7.31) can be used to obtain performance bounds for the estimation measures.

Theorem 7.5.3. *Under the settings of Theorem 7.5.2, the estimation quality losses compared to the case where all features in Θ_t are used satisfy*

$$\frac{\rho_v(\mathbf{H}_{t,T}(\Theta_t)) - \rho_v(\mathbf{H}_{t,T}(\Phi_t))}{\rho_v(\mathbf{H}_{t,T}(\Phi_t))} \leq \frac{4\bar{\chi}}{1-\epsilon} - 1 \tag{7.32}$$

$$\rho_e(\mathbf{H}_{t,T}(\Theta_t)) - \rho_e(\mathbf{H}_{t,T}(\Phi_t)) \leq n \log\left(\frac{4\bar{\chi}}{1-\epsilon}\right) \tag{7.33}$$

$$\frac{\rho_\lambda(\mathbf{H}_{t,T}(\Theta_t)) - \rho_\lambda(\mathbf{H}_{t,T}(\Phi_t))}{\rho_\lambda(\mathbf{H}_{t,T}(\Phi_t))} \leq \frac{4\bar{\chi}}{1-\epsilon} - 1, \tag{7.34}$$

with probability at least $1/4$.

7.5.4 Implementation of Algorithm

The nature of the performance guarantees provided in Theorem 7.5.2 and 7.5.3 motivates us to run the algorithm with multiple random seeds, i.e., by conducting Monte-Carlo simulations. To this end, we choose a design that corresponds to the minimal value of the estimation measure of interest. We inspect that there are (at least) two steps during the feature selection process that are amenable to parallel implementation: (i) evaluation of sampling probabilities π_f for different features, and (ii) independent executions of Algorithm 7 for the purpose of finding different designs.

7.5.5 Time-Complexity Analysis

To find the sampling probabilities, we need $O(N_t T^3)$ operations, where $N_t = |F_t|$ is the number of available features at time t. One execution of Algorithm 7 requires $O(qT^2)$ operations. Evaluation of any of these estimation measures requires $O(T^3)$ operations. Therefore, if we run p independent samples of this algorithm, we will need $O(pqT^2 + pT^3)$ operations. Hence, the overall feature selection will require $O(N_t T^3 + pqT^2 + pT^3)$ operations.

For comparison purposes, we also analyze the time complexity of feature selection using the greedy method of [170]. For this method, iteratively, we should examine all candidates and find the feature whose addition will enhance the estimation quality more than the remaining features. This method requires $O(qN_t T^3)$ operations. In the worst-case, $q = O(N_t)$. Thus, in the worst case, its time complexity is $O(N_t^2 T^3)$, i.e., quadratic in the

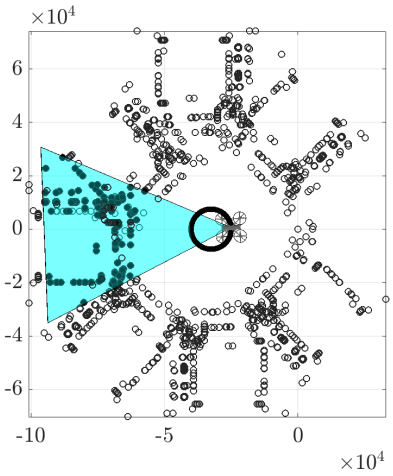

Figure 7.2: The top view of the navigation environment. The 3D reference curve is seen as a circle from this view.

number of available features. This suggests that the random sampling using the leverage scores can potentially be faster than the greedy method (see next section for a numerical example).

7.6 Simulation Results

We explain the details of a numerical experiment, which is conducted to demonstrate effectiveness of Algorithm 7.

7.6.1 Model and Environment Description

We consider a robot that is translating and rotating. Let us denote its position vector by $x_\tau^T = [p_\tau, y_\tau, z_\tau]$[1]. We suppose that the high-level dynamics of robot follow[2]

$$\begin{cases} p_{\tau+1} = p_\tau + u_\tau^p + \delta_\tau^p \\ y_{\tau+1} = y_\tau + u_\tau^y + \delta_\tau^y \\ z_{\tau+1} = z_\tau + u_\tau^z + \delta_\tau^z \end{cases}, \tag{7.35}$$

[1]We use letter p for the first coordinate to prevent conflict with use of position vector x.

[2]The implicit assumption here is that robot is already controlled by an internal feedback control mechanism and (7.35) represents the dynamics of the robot from reference (or tracking) signal to the state variables.

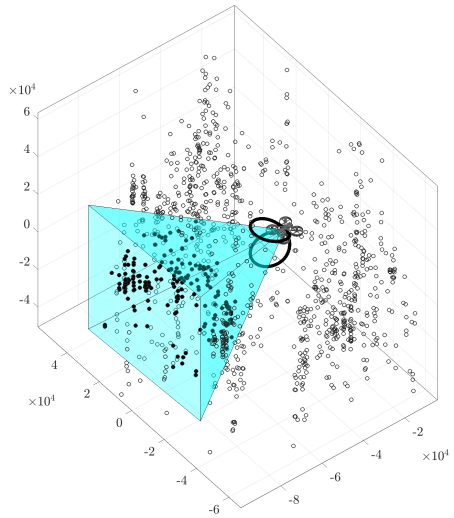

Figure 7.3: A snapshot of the environment. The blue pyramid demonstrates the camera's field of view. The features that are inside the frame at this time are highlighted. The deformed 8-shaped curve is the reference path as parametrized in (7.36) (see Fig. 7.2 for the top view as well). The robot is also rotating according to (7.38).

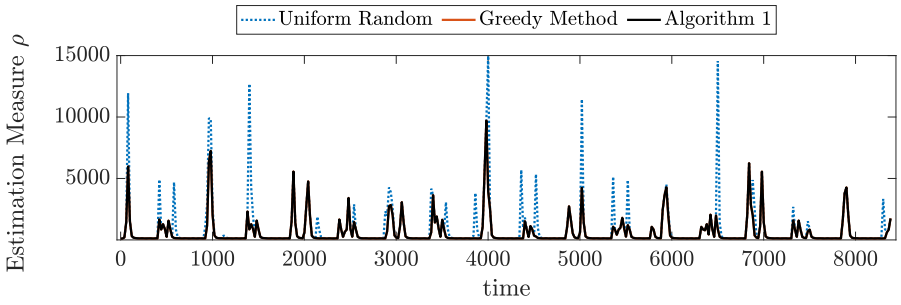

Figure 7.4: The estimation measure values resulting from three method. The curves corresponding to the to the greedy method and the proposed method may not be distinguished in this plot.

where $u_\tau^T := [u_\tau^p, u_\tau^y, u_\tau^z]$ is the input command signal and $\delta_\tau^T := [\delta_\tau^p, \delta_\tau^y, \delta_\tau^z]$ is the random process describing the uncertainty propagation due to the motion of the robot (compare to (7.7)). We consider the robot that is planning to move in the following reference path

$$\begin{cases} p_\tau^{\text{ref}} = p_0 + R\cos(\omega\tau) \\ y_\tau^{\text{ref}} = R\sin(\omega\tau) \\ z_\tau^{\text{ref}} = R\sin(\omega\tau/2) \end{cases}, \tag{7.36}$$

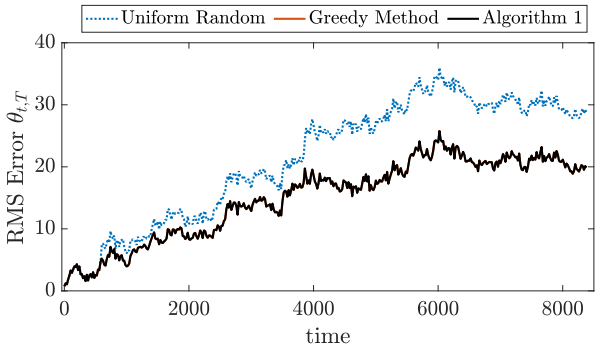

Figure 7.5: The RMS error in the positions of the robot resulting from feature selection using different methods. The curces corresponding to the greedy method and Algorithm 1 may not be distinguished in this plot.

which looks like a deformed 3-D number 8 (see Fig. 7.3). To set the control inputs, we set

$$\begin{cases} u_\tau^p = p_{\tau+1}^{\text{ref}} - \bar{\mu}_\tau^p \\ u_\tau^y = y_{\tau+1}^{\text{ref}} - \bar{\mu}_\tau^y \\ u_\tau^z = z_{\tau+1}^{\text{ref}} - \bar{\mu}_\tau^z \end{cases} . \tag{7.37}$$

Moreover, we suppose that the Euler angles describing the absolute orientation of the camera at time τ are given by

$$\begin{cases} \alpha_\tau = 2\pi \sin(\omega_r \tau) \\ \beta_\tau = -\dfrac{\pi}{2} + \dfrac{\pi}{20} \sin(\omega_r \tau) \\ \gamma_\tau = 0 \end{cases} , \tag{7.38}$$

where the sequence of rotations is z-y-p. The visible landmarks in the environments consists of 1752 points in the space, which is constructed by by putting a circular array of randomly sampled points in a 3-D model of a room[3]. Two views from a snapshot of the environment have been illustrated in Fig. 7.2 and Fig. 7.3. We set parameters $R = 7500$, $\omega = 0.08$, $\omega_r = 0.0064$, $\sigma = 0.1$ and $\Lambda_t = \text{diag}(4, 4, 16)$ and initial covariance to be $\Sigma_0 = I$.

[3]The STL graphical file is adapted from `https://grabcad.com/library/room-blender-test-1`

7.6.2 Different Approaches for Feature Selection

We consider the navigation setup for 400 time horizons each of length $T = 20$. The overall simulation consists of doing almost 107 full turns around the 3-D path of interest. For each horizon, after finding the eligible features to track (i.e., features with full rank information matrix $(\mathbf{E}^f_{t,T})^T\mathbf{E}^f_{t,T}$), we select at most half of the features. For each horizon, we do this task via three different methods:

(i) randomized sampling by leverage scores: we run Algorithm 7 for $p = 50$ independent experiments and choose the set Φ_t which induces the minimal value of the estimation measure. We denote the CPU time spent on this task by τ_t.

(ii) randomized sampling using uniform probabilities: we run Algorithm 7 for $p = 50$ independent experiments, except that instead of evaluating the probabilities, we assume an equal probability for each feature to be sampled. Similar to the previous case, we choose the design that induces minimal value of the estimation measure. We denote this design by Φ^u_t, where u stands for uniform probabilities. Similarly, we denote the corresponding CPU time for this task by τ^u_t.

(ii) greedy method: the features are added one-by-one, where at each iteration the feature which enhances the estimation quality the most is selected [170]. This method produces a single design for the feature selection denoted by Φ^g_t, where g stands for the greedy method.

In this example, we consider the estimation measure ρ_v as the monotone function governing the feature selection.

7.6.3 Metrics for Comparison of Methods

Many researchers have observed that the greedy method over-performs other approaches in several similar combinatorial problems [170, 172]. In general, the brute-force method in these settings is computationally infeasible[4], we select the greedy method as the base approach. Moreover, to have a clear understanding of the error in the position, we define

[4]For instance, for a choice of 25 features out of 50 candidates we have to examine more than 10^{14} possible combinations. In fact, finding the brute-force solution becomes rapidly computationally prohibitive as the size of the candidate set grows.

the root mean squared error (RMSE) as we define

$$\theta_{t,T} := \frac{1}{3(T+1)} \sqrt{\sum_{\tau=t}^{t+T} \|x_\tau - \mu_t\|_2^2}, \tag{7.39}$$

for $t \in \{0, T, 2T, \dots\}$. We define similar error indices for the uniform random (method *(ii)*) and the greedy method (method *(iii)*) as well and denote them by $\theta^u_{t,T}$ and $\theta^g_{t,T}$, respectively. To compare the relative difference of these values, we use

$$\phi_{t,T} := \frac{\theta_{t,T} - \theta^g_{t,T}}{\theta^g_{t,T}} \times 100. \tag{7.40}$$

Similarly, we define $\phi^u_{t,T}$, which compares the value of RMSE resulting from the totally random choice of feature with greedy selection. Finally, to compare the CPU times, we look at

$$\kappa_t := \frac{\tau_t}{\tau^g_t}, \tag{7.41}$$

which represent the ratio of the CPU time spent in methods *(i)* to the one spent by the greedy method. Similarly, we use κ^u_t to compare the time spent by method *(ii)* with method *(iii)*.

7.6.4 Numerical Results

In Fig 7.4, we show the resulting values of the estimation measure versus time, which demonstrate that the estimation measure resulting from Algorithm 7 is almost identical to the estimation measure resulting from the greedy method. This is not the case for the totally random choice of features, in which larger spikes can be observed.

In Fig. 7.5, we illustrate the values of the RMS error versus times for these methods. The errors corresponding to the feature selection using Algorithm 7 is very close to those values for the greedy method, while the totally random method may result in larger errors. This shows that in this example, not only the choice of features is a non-trivial computational task, but also our algorithm functions with quality and reliability that is very close to these

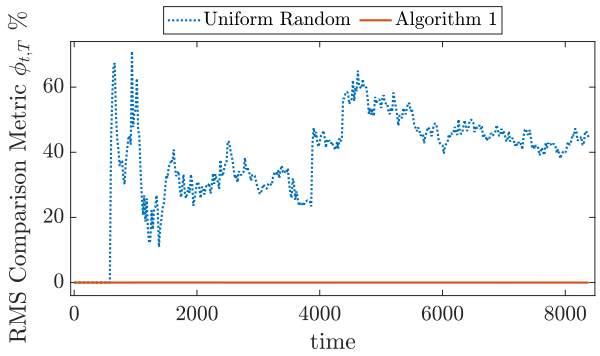

Figure 7.6: Comparison of the RMS error resulting from the uniform random method and our proposed approach with the greedy method. This plot shows a speed-up by almost an order of magnitude in the feature selection using Algorithm 7.

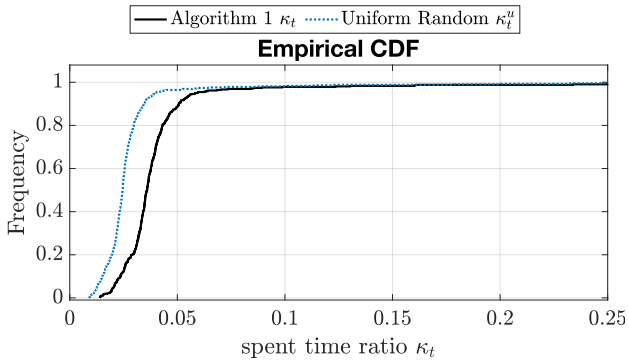

Figure 7.7: The empirical CDF's for parameters κ_t and κ_t^u, which show the ratio of the spent CPU time for the random sampling (including all independent 50 experiments) to the greedy method. The plot show that in this example the random sampling is faster than the greedy method by almost an order of magnitude.

factors in the case of the greedy method. The metric $\phi_{t,T}$ is also illustrated in Fig. 7.6, which quantifies these deviations.

Finally, we compare the CPU times spent on each method. In Fig. 7.7, we demonstrate the experimental CDF's of the parameters κ_t and κ_t^u, which show that the randomized methods are considerably faster than the greedy method in most cases. For instance, these data suggest that Algorithm 7 has been more than 20 times faster than the greedy method in about 85% of the assigned tasks, while in most cases it has been least 10 times faster. Note that the time includes running the random sampling algorithms for 50 independent experiments

7.7 Discussion and Conclusion

We propose a randomized algorithm for visual feature selection over a fixed-length moving time-horizon. The idea is to associate a sampling probability to each candidate feature, randomly sample a subset of features according to these probabilities for a number of independent experiments, and select the outcome with the best estimation quality. The most important property of our algorithm is that its time complexity scales linearly with the number of features, which makes it suitable for applications with hundreds or thousands of features.

If the estimation measure enjoys submodularity, then the greedy method provides a performance guarantee compared to the optimal solution [170, 173]. However, it is known that certain measures, for instance, ρ_v, are not submodular [143]. Moreover, in our work, we offer a different type of performance guarantee. Theorem 7.5.3 compares the estimation quality to the case that we leverage all features for tracking. Nevertheless, our extensive numerical simulations assert that the resulting estimation quality from our algorithm and that of the greedy method are often close to each (e.g., see Fig. 7.4). Further research is required to uncover the practical and theoretical differences of this randomized algorithm and the greedy method.

According to the time complexity analysis in Section 7.5.5, if the number of selected features is small, the computational cost of our randomized method will be comparable to those of greedy methods. However, in the worst case, the greedy-method scales quadratically with the number of candidate features (i.e., scaling with $N_t^2 = |\Theta_t|^2$). This justifies the significant speed-up in the feature-selection that is observed in our numerical simulations (see Fig. 7.7). The low time complexity of our method opens up new opportunities for real-time implementation of this algorithm and utilizing it for agile robot navigation.

Appendix

Appendix A: Proof of Lemma 7.3.1

After one episode of motion in the horizon, one can verify that the updates to covariance and the mean are given by

$$\bar{\Sigma}_{t+1} = A_{t+1}\bar{\Sigma}_t A_{t+1}^T + \Lambda_{t+1}, \tag{7.42}$$

$$\bar{\mu}_{t+1} = \bar{\Delta}_{t+1}, \tag{7.43}$$

respectively. Because these update laws also work for any time instant in the horizon, we can use the similar update

$$\bar{\Sigma}_{\tau} = A_{\tau}\bar{\Sigma}_{\tau-1} A_{\tau}^T + \Lambda_{\tau}, \tag{7.44}$$

$$\mu_{\tau} = \bar{\Delta}_{\tau}, \tag{7.45}$$

for every instant $\tau \in \{t+1, \ldots, t+T\}$. To fully identify covariance matrix $\bar{\mathbf{\Sigma}}_{t,T}$, we can show that for all $\tau_1, \tau_2 \in \{t, t+1, \ldots, t+T\}$, with $\tau_1 < \tau_2$,

$$\mathbb{E}\{x_{\tau_1} x_{\tau_2}^T\} - \mathbb{E}\{x_{\tau_1}\}\mathbb{E}\{x_{\tau_2}^T\} = \left(\prod_{i=1}^{\tau_2-\tau_1} A_{\tau_2-i-1}\right)\bar{\Sigma}_{\tau_1} := \bar{\Sigma}_{\tau_1,\tau_2}.$$

The proof of this relationship can be conducted by induction. Then, we observe that the covariance matrix has the structure

$$\bar{\mathbf{\Sigma}}_{t,T} = \begin{bmatrix} \bar{\Sigma}_t & \bar{\Sigma}_{t,t+1} & \cdots & \bar{\Sigma}_{t,t+T} \\ \bar{\Sigma}_{t,t+1}^T & \bar{\Sigma}_{\tau+1} & \cdots & \bar{\Sigma}_{t+1,t+T} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\Sigma}_{t,t+T}^T & \bar{\Sigma}_{t+1,t+T}^T & \cdots & \bar{\Sigma}_{t,T} \end{bmatrix}, \tag{7.46}$$

where the diagonal and off-diagonal matrix terms are supposed to be evaluated by iterative application of (7.44) and (7.46), respectively. For the mean, the update is simply achieved by stacking the updated mean variables in (7.43). This completes the proof.

Appendix B: Proof of Lemma 7.3.2

The proof of this lemma is based on an approach similar to one given in [170]. Note that initial information matrix of the stacked variable $[\mathbf{x}_{t,T}^T \; y_f^T]^T$ is

$$\begin{bmatrix} \bar{\mathbf{H}}_{t,T} & 0 \\ 0 & 0 \end{bmatrix},$$

as we assume no information regarding the location of the feature f[5]. Then, the update to the information matrix based on the observation model is of the form

$$\widetilde{\mathbf{H}}_{t,T} := \begin{bmatrix} \bar{\mathbf{H}}_{t,T} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} (\mathbf{F}_{t,T}^f)^T \mathbf{F}_{t,T}^f & (\mathbf{F}_{t,T}^f)^T \mathbf{E}_{t,T}^f \\ (\mathbf{E}_{t,T}^f)^T \mathbf{F}_{t,T}^f & (\mathbf{E}_{t,T}^f)^T \mathbf{E}_{t,T}^f \end{bmatrix},$$

(e.g. see [171] to see its derivation based on the Bayes law). Given the structure of this matrix, using the Schur complement, one can show that

$$(\tilde{\mathbf{H}}_{t,T})^{-1} = \begin{bmatrix} \mathbf{H}_{t,T}^{-1} & -\mathbf{H}_{t,T}^{-1} \mathbf{C}_{t,T}^f \\ \star_1 & \star_2 \end{bmatrix},$$

where the value of the starred matrices will not be needed and we have

$$\mathbf{H}_{t,T} = \bar{\mathbf{H}}_{t,T} + \mathbf{H}_{t,T}^f \tag{7.47}$$

$$\mathbf{C}_{t,T}^f := (\mathbf{F}_{t,T}^f)^T \mathbf{E}_{t,T}^f \big((\mathbf{E}_{t,T}^f)^T \mathbf{E}_{t,T}^f \big)^{-1}. \tag{7.48}$$

Now, note that the update to the information vector of the stacked variable $[\mathbf{x}_{t,T}^T \; y_f^T]^T$ is of the form

$$\widetilde{\mathbf{b}}_{t,T}^T := \begin{bmatrix} \bar{\mathbf{b}}_{t,T}^T \\ 0 \end{bmatrix} + \begin{bmatrix} (\mathbf{F}_{t,T}^f)^T \mathbf{z}_{t,T}^f \\ (\mathbf{E}_{t,T}^f)^T \mathbf{z}_{t,T}^f \end{bmatrix}.$$

[5] Informally, this is due to the fact that we are not building a map out of these observations. Instead, we only track them to enhance the quality of our estimation from the position of the robot.

Therefore, the update for the mean of the stacked variable is given by

$$\begin{aligned}\widetilde{\boldsymbol{\mu}}_{t,T} &= (\tilde{\mathbf{H}}_{t,T})^{-1}\widetilde{\mathbf{b}}_{t,T}^{T} \\ &= \begin{bmatrix} \mathbf{H}_{t,T}^{-1}\big(\bar{\mathbf{b}}_{t,T}^{T} + (\mathbf{F}_{t,T}^{f})^{T} - \mathbf{C}_{t,T}^{f}(\mathbf{E}_{t,T}^{f})^{T}\big)\mathbf{z}_{t,T}^{f}) \\ \star_3 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{H}_{t,T}^{-1}(\bar{\mathbf{b}}_{t,T}^{T} + \mathbf{B}_{t,T}^{f}\mathbf{z}_{t,T}^{f}) \\ \star_3 \end{bmatrix},\end{aligned}$$

where in the last one we have used the definition of $\mathbf{B}_{t,T}^{f}$ and the starred element will not be used. Based on the partition of the stacked variable, we inspect that

$$\boldsymbol{\mu}_{t,T} = \mathbf{H}_{t,T}^{-1}(\bar{\mathbf{b}}_{t,T} + \mathbf{B}_{t,T}^{f}\mathbf{z}_{t,T}^{f}). \tag{7.49}$$

Hence, based on the definition of the information vector, by inspection, we get that

$$\mathbf{b}_{t,T}^{T} = \bar{\mathbf{b}}_{t,T}^{T} + \mathbf{B}_{t,T}^{f}\mathbf{z}_{t,T}^{f}. \tag{7.50}$$

This completes the proof of the lemma.

Appendix C: Proof of Theorem 7.5.2

For all features $f \in \Theta_t$, the corresponding matrix $\bar{\mathbf{H}}_{t,T}^{f}$ can be decomposed as

$$\bar{\mathbf{H}}_{t,T}^{f} = \sum_{i=1}^{n} \mathbf{h}_{t,T}^{f,i}(\mathbf{h}_{t,T}^{f,i})^{T} := \sum_{i=1}^{n} \bar{\mathbf{H}}_{t,T}^{f,i}, \tag{7.51}$$

where vectors $\mathbf{h}_{t,T}^{f,i} \in \mathbb{R}^n$ are orthogonal to each other for each $\bar{\mathbf{H}}_{t,T}^{f}$. Moreover, we define the refined leverage scores as follows.

Definition 7.7.1. *For a given set of features Θ_t and decomposition of information matrices given in* (7.51)*, the refined leverage scores are nonnegative numbers defined by*

$$r_{fi} := \mathrm{Tr}\left(\mathbf{H}_{t,T}(\Theta_t)^{-1}\bar{\mathbf{H}}_{t,T}^{f,i}\right). \tag{7.52}$$

Algorithm 8 Randomized Feature Selection (Analysis Version with Additional Lines)

input: initial information matrix $\bar{\mathbf{H}}_{t,T}$
set of available features Θ_t, number of samples q
output: selected features Φ_t, information matrix $\mathbf{H}_{t,T}$
set of indices $\hat{\Phi}_t$, weight function w_t ▷

initialize: $\hat{\Phi}_t = \varnothing$, $\mathbf{H}_{t,T} = \bar{\mathbf{H}}_{t,T}$
$w_t(.,.) = 0$ ▷
for $k = 1$ to q **do**
sample a feature from Θ_t using distribution $\pi \to f$
select the corresponding update matrix

$$\mathbf{H} \leftarrow \mathbf{H}_{t,T}^{f}$$

sample from 1 to n with distribution $p_f \to i$ ▷
update weight function: ▷

$$w_t(f,i) \leftarrow w_t(f,i) + (q\pi_{fi})^{-1}$$ ▷

if $f \notin \Phi_t$, **then**
add f to Φ_t
update the information matrix:

$$\mathbf{H}_{t,T} \leftarrow \mathbf{H}_{t,T} + \mathbf{H}$$

end if
if $(f,i) \notin \hat{\Phi}_t$, **then** ▷
add (f,i) to $\hat{\Phi}_t$ ▷
end if ▷
end for

for every $f \in \Theta_t$ *and* $i \in \{1, \ldots, n\}$.

Similarly, let us denote the refined sampling probabilities by $\pi_{fi} := r_{fi}/n$. Due to the orthogonality of the vectors $\mathbf{h}_{t,T}^{f,i}$, one inspects that

$$r_f = \sum_{i=1}^{n} r_{fi}, \quad \pi_f = \sum_{i=1}^{n} \pi_{fi}, \tag{7.53}$$

for $f \in \Theta_t$. We define

$$p_f(i) = p_{fi} = \frac{\pi_{fi}}{\sum_{i=1}^{n} \pi_{fi}} = \frac{\pi_{fi}}{\pi_f}.$$

We observe that over each $f \in \Theta_t$, numbers p_{fi} also constitute a probability distribution (i.e, they sum up to 1). Now, we modify Algorithm 7 to get Algorithm 8, wherein lines

labeled with symbol $\triangleright$ are only required for performance analysis and do not need to be conducted during the execution of the algorithm; i.e., removing those lines from Algorithm 8 will give us Algorithm 7. Algorithm 8 assigns values to a weight function that lets us assess the quality of estimation. The weight function $w_t : \hat{\Phi}_t \to \mathbb{R}_+$ is a bounded random variable, where $w_t(f,i)$ may take different realizations drawn from $\big\{p(q\pi_{fi})^{-1} \;\big|\; p = 0, 1, \ldots, q\big\}$.

Now, we build up our proof based on some of the steps taken in the proof of Theorem 5 in [57] and steps from [156], wherein the authors prove a similar result for the case that selected matrices are rank-one. Using the leverage scores (7.29), one can verify that because $\bar{\mathbf{H}}_{t,T}^{f,i}$ for each $(f,i) \in \hat{\Phi}_t$ is rank-one, then by applying similar steps to the proof of Theorem 5 in [57], with probability at least $1/2$ we have

$$(1-\epsilon)\mathbf{H}_{t,T}(\Theta_t) \preceq \mathbf{H}_w, \tag{7.54}$$

where $\mathbf{H}_w$ is given by

$$\mathbf{H}_w = \sum_{(f,i)\in\hat{\Phi}_t} w_t(f,i)\,\bar{\mathbf{H}}_{t,T}^{f,i} = \sum_{(f,i)\in\hat{\Phi}_t} \frac{\phi_{fi}}{q\pi(f)}\,\bar{\mathbf{H}}_{t,T}^{f,i}, \tag{7.55}$$

and $\phi_{fi} \geq 0$ is the frequency of the times that feature f and then index $i \in \{1, \ldots, n\}$ are sampled by Algorithm 8. Now, consider the matrices

$$T := \Big[(\mathbf{h}_{t,T}^{f,i})^T\Big]_{(f,i)\in\hat{\Phi}_t}, \quad W = \operatorname{diag}\big(w_t(f,i)\big)\big|_{(f,i)\in\hat{\Phi}_t}.$$

Let us introduce an *artificial* linear parameter estimation problem based on the model

$$y = T\theta + \eta \tag{7.56}$$

where observation is given by y and η is a zero mean Gaussian measurement noise of independent components and covariance $\mathbb{E}\left\{\eta\eta^T\right\} = I$. In the next step, let us consider

estimators $\hat{\theta}$ and $\tilde{\theta}$ that are given by

$$\hat{\theta} = \left(T^T T\right)^{-1} T^T y. \tag{7.57}$$

$$\tilde{\theta} = \left(T^T W T\right)^{-1} T^T W y. \tag{7.58}$$

It is straightforward to verify that both of them are unbiased estimators for θ. Since covariance of noise is $\mathbb{E}\left\{\eta\eta^T\right\} = I$, the unweighted least-squares estimator is the optimal estimator. Consequently, by Gauss-Markov theorem

$$\mathbb{E}\left\{\tilde{\theta}\tilde{\theta}^T\right\} \succeq \mathbb{E}\left\{\hat{\theta}\hat{\theta}^T\right\}. \tag{7.59}$$

Now, we explicitly write down the two sides of (7.59). First observe that

$$\mathbb{E}\left\{\hat{\theta}\hat{\theta}^T\right\} = \left(T^T T\right)^{-1} = \left(\sum_{(f,i)\in\hat{\Phi}_t} \bar{\mathbf{H}}_{t,T}^{f,i}\right)^{-1} := \hat{\mathbf{H}}_{t,T}^{-1}. \tag{7.60}$$

Because $\mathbf{H}_w = T^T W T$, we can also write

$$\mathbb{E}\left\{\tilde{\theta}\tilde{\theta}^T\right\} = \mathbf{H}_w^{-1} T^T W^2 T \mathbf{H}_w^{-1}. \tag{7.61}$$

We define χ to be a random variable given by

$$\chi = \inf\left\{\gamma > 0 \,\middle|\, \sum_{(f,i)\in\hat{\Phi}_t} w_t(f,i)\left(\gamma - w_t(f,i)\right)\bar{\mathbf{H}}_{t,T}^{f,i} \succeq \mathbf{0}\right\}.$$

Moreover, we set $\bar{\chi} := \mathbb{E}\{\chi\}$. Let us isolate the term in the middle of (7.61). Based on the definition of random variable χ, one can write

$$T^T W^2 T \preceq \chi \mathbf{H}_w \tag{7.62}$$

One can show that for every three positive-definite matrices X_1, X_2, X_3 with $X_1 \succeq X_2$,

inequality

$$X_3 X_1 X_3 \succeq X_3 X_2 X_3 \tag{7.63}$$

holds. Combining (7.61), (7.62), and (7.63) we find that

$$\mathbb{E}\left\{\tilde{\theta}\tilde{\theta}^T\right\} \leq \mathbf{H}_w^{-1}(\chi \mathbf{H}_w)\mathbf{H}_w^{-1} = \chi \mathbf{H}_w^{-1}. \tag{7.64}$$

Based on Markov inequality, it holds that

$$\chi \leq \frac{1}{1-\frac{3}{4}}\mathbb{E}\{\chi\} = 4\bar{\chi} \tag{7.65}$$

with probability at least 3/4. From (7.64) and (7.65), we get

$$\mathbb{E}\left\{\tilde{\theta}\tilde{\theta}^T\right\} \preceq 4\bar{\chi}\mathbf{H}_w. \tag{7.66}$$

On the other hand, because (7.54) holds, with probability at least 1/2 By taking inverse, we get

$$\mathbf{H}_w \preceq (1-\epsilon)^{-1}\mathbf{H}_{t,T}(\Theta_t)^{-1}, \tag{7.67}$$

If the event described in (7.65) is denoted by $\mathfrak{A}$ and the event described by (7.67) is denoted by $\mathfrak{B}$, then

$$\mathbb{P}(\mathfrak{A} \cap \mathfrak{B}) = \mathbb{P}(\mathfrak{A}) + \mathbb{P}(\mathfrak{B}) - \mathbb{P}(\mathfrak{A} \cup \mathfrak{B}) \geq \frac{3}{4} + \frac{1}{2} - 1 = \frac{1}{4}.$$

Therefore both (7.65) and (7.67) hold with probability at least 1/4; i.e, the inequality

$$\mathbb{E}\left\{\tilde{\theta}\tilde{\theta}^T\right\} \preceq 4\bar{\chi}(1-\epsilon)^{-1}\mathbf{H}_{t,T}(\Theta_t)^{-1},$$

holds with probability at least 1/4. Because (7.59) holds, with probability at least 1/4

$$\hat{\mathbf{H}}_{t,T}^{-1} \preceq 4\bar{\chi}(1-\epsilon)^{-1}\mathbf{H}_{t,T}(\Theta_t)^{-1}. \tag{7.68}$$

Or equivalently,

$$\hat{\mathbf{H}}_t \succeq \frac{1-\epsilon}{4\bar{\chi}} \mathbf{H}_{t,T}(\Theta_t). \tag{7.69}$$

As the final step, we observe that

$$\begin{aligned} \mathbf{H}_{t,T}(\Phi_t) &= \bar{\mathbf{H}}_{t,T} + \sum_{f \in \Phi_t} \mathbf{H}^f_{t,T} \\ &\succeq \frac{|\Phi_t|}{N_t} \bar{\mathbf{H}}_{t,T} + \sum_{f \in \Phi_t} \mathbf{H}^f_{t,T} \\ &= \sum_{f \in \Phi_t} \frac{1}{N_t} \bar{\mathbf{H}}_{t,T} + \mathbf{H}^f_{t,T} = \sum_{f \in \Phi_t} \bar{\mathbf{H}}^f_{t,T} \\ &\succeq \sum_{(f,i) \in \hat{\Phi}_t} \bar{\mathbf{H}}^{f,i}_{t,T} = \hat{\mathbf{H}}_t. \end{aligned} \tag{7.70}$$

Combining (7.69) and (7.70) we conclude that

$$\mathbf{H}_{t,T}(\Phi_t) \succeq \frac{1-\epsilon}{4\bar{\chi}} \mathbf{H}_{t,T}(\Theta_t), \tag{7.71}$$

with a probability that exceeds $1/4$

Appendix D: Proof of Theorem 7.5.3

Combining (7.31) and the definition of ρ_v, we find that the following inequality with probability at least $1/4$ holds.

$$\rho_v(\mathbf{H}_{t,T}(\Phi_t)) \leq \frac{4\bar{\chi}}{1-\epsilon} \operatorname{Tr}(\mathbf{\Sigma}_{t,T}(\Theta_t)) \tag{7.72}$$

$$= \frac{4\bar{\chi}}{1-\epsilon} \rho_v(\mathbf{H}_{t,T}(\Theta_t)). \tag{7.73}$$

The proof of (7.34) is similar. For the entropy estimation measure, combining (7.31) and definition of ρ_e, we find that the following inequality with probability at least $1/4$ holds.

$$\begin{aligned}\rho_e(\mathbf{H}_{t,T}(\Phi_t)) &= \log(\det(\mathbf{H}_{t,T}(\Phi_t)^{-1})) \\ &\leq \log\left(\det\left(\frac{4\bar{\chi}}{1-\epsilon}\mathbf{H}_{t,T}(\Theta_t)^{-1}\right)\right) \\ &= n\log\left(\frac{4\bar{\chi}}{1-\epsilon}\right) + \rho_e(\mathbf{H}_{t,T}(\Theta_t)).\end{aligned}$$

Part III:

Problems in Reinforcement Learning

Chapter 8

Multi-Agent Image Classification via Reinforcement Learning

8.1 Introduction

With the rising interest in the Internet of Things (IoT), the demand for design of autonomous agents that are capable of cooperation is increasing.. The interconnected robots will be major players in the future, accomplishing many duties in industrial automation [174], military support [175], and health-care [176]. In many of these applications, a major issue is that every agent has limited sensing capabilities, and therefore, may not have sufficient information for accomplishing a complex task. One way to mitigate this shortcoming is to let the task to be solved collectively by multiple agents. In the context of machine learning, this means that the agents need not only to learn through individual interaction with their environment but also they can learn from each others' experiences through communication.

In this Chapter, we study the multi-agent image classification problem within an unknown environment. The setup consists of an environment, in which multiple homogeneous agents, each with a partial observation of the environment, are collaborating to do a classification task. To explore the environment efficiently, the agents need to learn how to optimally traverse the environment. The agents receive new observations from the environ-

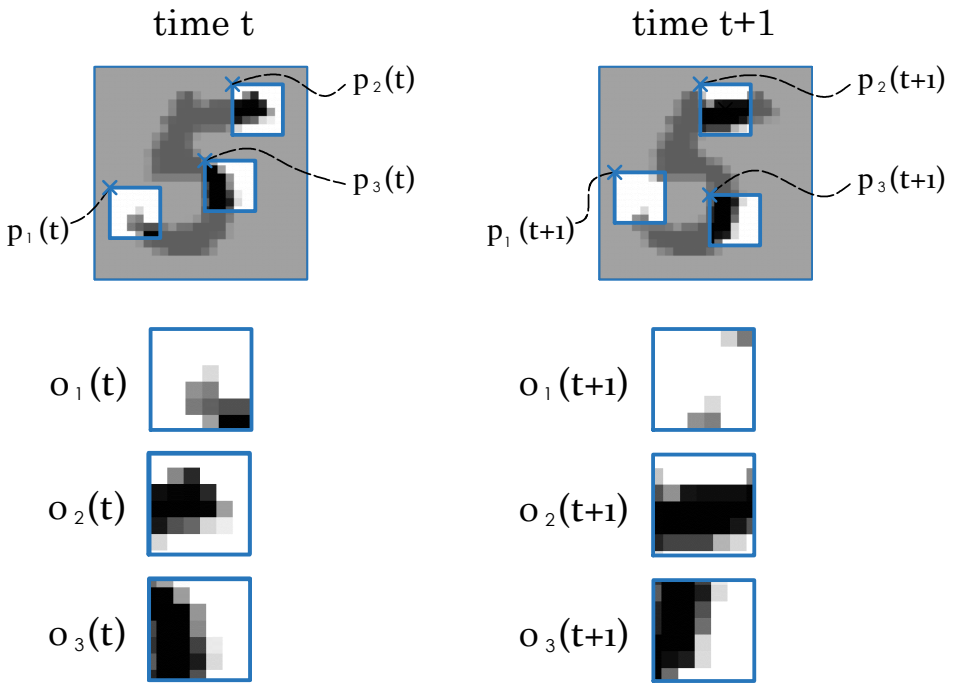

Figure 8.1: An example showing how the spatial variables dictate the observations of agents. The observations by 3 agents at two consecutive time instants have been magnified. This highlights the need for a communication mechanism and temporal memories.

ment through re-positioning. Moreover, they are capable of communication with each other to update their beliefs. This is motivated by the idea that a more experienced teammate could provide or explain hints, which can be used to facilitate the perception. We are interested in maximizing a long-term collective objective such that the agents may effectively coordinate and cooperate to correctly classify the environment. Our goal in this work is to co-design the decentralized data-processing, communication, and decision-making policies on the agents, while that the agents are capable of establishing autonomous connections to other agents according to an information structure (see Fig. 8.1 for an example).

For solving this problem, we formulate it within a multi-agent reinforcement learning framework and propose a policy gradient, which can effectively optimize the agents' behavior. In contrast to the vanilla policy gradient settings, our framework introduces a differentiable reward rather instead of a reward that is independent of the network parameters. The proposed mechanism enables the policy gradient algorithm to not only maximize the probability of generating desirable outcomes, but also to explicitly increase the rewards.

The mathematical derivation of the latter argument is given in Section 8.4, which generalizes the policy gradient approach for the case of differentiable rewards.

We have demonstrated a descent performance of the proposed framework in the MNIST dataset of handwritten digits. We can correctly classify 88% of the testing dataset by using two agents, each only with 2 $\times$2 pixels observations. We observe that with larger observations or longer communications, the prediction quality further increases up to 97.75%.

8.2 Problem Statement

Suppose that N identical agents are in a static unknown environment. Let the agents start from a pre-determined spatial configuration. At each time step, each agent is capable of collecting a partial observation from the environment, performing some local data processing, and communicating the result with neighboring agents. The agents are allowed to communicate over a directed graph, where neighbors of an agent are those agents whose messages can be received by that agent. We assume that each agent knows its own pose with respect to the environment and can take certain actions to move and update its pose at each time step.

The collective objective of these agents is to classify the instance of the environment from a finite number of possibilities $\{1, 2, \ldots, M\}$ over a finite time horizon.

In order to decentralize the process throughout the execution, the actions by each agent should be decided solely based on the local information available to them. As a result, agents need to learn how to communicate, extract relevant features and specifications from partial observations, navigate in the environment, and reliably solve the classification problem.

In Section 8.4, we propose a modular architecture for the network of multiple agents and discuss the details of different modules of an agent to achieve decentralized classification. In Section 8.6, we demonstrate the required steps for learning the design parameters using reinforcement learning techniques.

8.3 Connections to the Literature

In most of the multi-agent reinforcement learning literature, it is typical to assume that the agents are non-identical since their respective policies might be very distinct [177,178]. Even though this assumption might be non-avoidable in some applications, assuming the existence of a common shared policy for all agents can be helpful, especially when their policies can be distinguished via some visible characteristics e.g. their location. Among the literature, [179] has considered a similar setup for the homogeneity of the agents. This assumption provides us with the advantage that the agents can learn from each others' experience. A similar idea is followed in [180], where an agent interacts with multiple instances of an environment, allowing her to learn from a concurrent stream of experiences. Even though they study a single-agent scenario, the homogeneity assumption allows approaching this multi-agent problem as a single-learner problem.

In series of works on multi-agent reinforcement learning, the focus is on cases where the data is spread over a number of agents and the goal is to conduct the *training* in a distributed or decentralized manner. For instance, a promising framework for this purpose is the Federated Learning, where the goal is to conduct the stochastic gradient descent by combining partially computed gradients over different agents [181–183]. Contrary to this line of research, we consider settings in which the agents need to communicate throughout the *execution* as well.

In a more related paper, the authors of [178] consider a value function approximation approach for decentralized learning with interconnected agents and bring operational guarantees in the case of linear value functions. A similar multi-agent learning problem is addressed in [184], where deep neural networks are used as function approximators. A generalization of this problem within an actor-critic scenario appears in [185]. Contrary to these works, we address the case where each data point is observed by a number of agents that are collaborating to fulfill a task (e.g. see Fig. 8.1), while the next set of observations are affected by (locally) decided actions of the agents. Moreover, in our settings, the reward for reinforcement learning becomes differentiable. This necessitates revisiting the derivation of the policy gradients.

Another related line of research concerns the end-to-end design of distributed or decentralized control architectures, where the goal is to learn optimal control laws in a setting with multiple dynamic agents [186–188].

8.3.1 Notations and Preliminaries

The set of real numbers, nonnegative real numbers, and nonnegative integer numbers are denoted by $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{Z}_+$, respectively. Other sets are denoted by script letters; e.g. $\mathcal{A}$ while their cardinality is denoted by $|\mathcal{A}|$. We use bold letters to denote maps; e.g. $\mathbf{g}(x)$. The trainable parameter of a map is denoted by θ_i appearing as a subscript; e.g. $\mathbf{f}_{\theta_1}$. Map $\text{softmax}(x) : \mathbb{R}^M \to \mathbb{R}_+^M$ is standard softmax with k'th element given by $\exp(x_k)/\sum_{i=1}^{M} \exp(x_i)$. A directed graph $\mathcal{G}$ is characterized by a set of nodes (or vertices) $\mathcal{V} := \{1, 2, \ldots, N\}$ and a set of directed edges (or arcs) denoted as $\mathcal{E} \subset \{(i, j) : \ i, j \in \mathcal{V}, \ i \neq j\}$. We say node $j \in \mathcal{V}$ is an in-neighbor of node $i \in \mathcal{V}$ if $(j, i) \in \mathcal{E}$, and denote the set of all of its in-neighbors by $\mathcal{N}_i := \{j \in \mathcal{V} : \ (j, i) \in \mathcal{E}\}$.

8.4 Architecture of the Multi-Agent Network

We discuss details of a modular design to solve the image classification problem using multiple autonomous agents.

8.4.1 Temporal Evolution of Agents' Beliefs

In order to enable learning long-term dependencies during the classification task, we equipped each agent by a dynamic module using a Long Short-Term Memory (LSTM) cell [189]. The role of this module is to encapsulate the aggregate belief of an agent throughout the task. Following the widely accepted terminology [190], let us denote the hidden state and cell state of the LSTM module on agent $i \in \mathcal{V}$ at time $t \geq 0$ by $h_i(t) \in \mathbb{R}^n$ and $c_i(t) \in \mathbb{R}^n$, respectively. Each agent updates its own belief upon receiving new observations, communicating with its neighbors, and forming an information input $u_i(t) \in \mathbb{R}^{3n}$ that contains three components: features of local observations, the average of the decoded messages received from its neighbors, and information about its location. The time evolution of the belief

LSTM module is governed by

$$\begin{bmatrix} h_i(t+1) \\ c_i(t+1) \end{bmatrix} = \mathbf{f}_{\theta_1} \left(\begin{bmatrix} h_i(t) \\ c_i(t) \end{bmatrix}, u_i(t) \right), \tag{8.1}$$

where nonlinear map $\mathbf{f}_{\theta_1} : \mathbb{R}^{2n} \times \mathbb{R}^{3n} \to \mathbb{R}^{2n}$ is parametrized by a trainable vector $\theta_1 \in \mathbb{R}^{n_f}$. In the following subsections, we discuss each component of the information input to the LSTM module.

8.4.2 Agent Motion and Stochastic Action Policy

Let us represent the spatial state (or pose) of agent $i \in \mathcal{V}$ by $p_i(t) \in \mathbb{R}^d$ and the finite set of all possible actions, which agents can take, by $\mathcal{A}$. Each agent moves in the spatial domain according to dynamics

$$p_i(t+1) = \mathbf{g}\big(p_i(t), a_i(t+1)\big), \tag{8.2}$$

where $\mathbf{g} : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^d$ is a known transition map and action $a_i(t+1)$ is sampled from set $\mathcal{A}$ according to a probability mass function $\pi : \mathcal{A} \to \mathbb{R}$ that is computed as follows. We use a state-dependent stochastic action policy by updating the action probabilities according to

$$\pi(a \,|\, O(t)) = \boldsymbol{\pi}_{\theta_3}(a, \hat{h}_i(t+1)), \tag{8.3}$$

where a is an action in $\mathcal{A}$, $O(t)$ is the history of all observations by all agents at time, $\hat{h}_i(t+1)$ is the hidden state of the decision LSTM unit whose dynamics are governed by

$$\begin{bmatrix} \hat{h}_i(t+1) \\ \hat{c}_i(t+1) \end{bmatrix} = \mathbf{f}_{\theta_2} \left(\begin{bmatrix} \hat{h}_i(t) \\ \hat{c}_i(t) \end{bmatrix}, u_i(t) \right). \tag{8.4}$$

This LSTM unit is fed with exactly the same information input $u_i(t)$ as the belief LSTM module (8.1).

For the input to map $\boldsymbol{\pi}$, we consider one fully connected layer with a ReLU activation and one fully connected linear layer for the output.

Example 8.4.1. For a flying robot that can translate and rotate in the 3D space, a natural choice for spatial state $p_i(t)$ is a vector in $\mathbb{R}^6$ created by stacking three position components of the robot and three Euler angles describing its orientation (relative to the environment frame).

8.4.3 Inter-Agent Communication Architecture

The agents are allowed to communicate over a directed graph $\mathcal{G}$ with node set $\mathcal{V}$ and arc set $\mathcal{E}$. For distinct agents $i, j \in \mathcal{V}$, $(i, j) \in \mathcal{E}$ implies that agent j receives messages from agent i. Each agent generates a message[1] using its belief hidden state according to

$$m_i(t) = \mathbf{m}_{\theta_4}(h_i(t)), \tag{8.5}$$

where map $\mathbf{m}_{\theta_4} : \mathbb{R}^n \to \mathbb{R}^{n_m}$ is parameterized by a trainable vector $\theta_4 \in \mathbb{R}^{n_e}$. A sequence of two layers for is considered for this map: a fully-connected layer with ReLU activation followed by a fully connected linear layer for the output.

8.4.4 Observation Model and Feature Extraction

Suppose that agent i at time t collects (partial) observation $o_i(t) \in \mathbb{R}^{f \times f}$. It is assumed that agents' observations can be completely characterized by its pose $p_i(t)$. Thus,

$$o_i(t) = \mathbf{o}(I, p_i(t)), \tag{8.6}$$

where $I \in \mathbb{R}^{n_I \times n_I}$ is the entire image. This identity can be interpreted as the *repeatability* property of the observations: two agents with different past history will observe the same image provided that they both have identical poses at the observation time. The relevant

[1] It is assumed that when agent i broadcasts its message $m_i(t) \in \mathbb{R}^{n_m}$ at time t, all neighboring agents receive identical copies of that message.

features of an observation can be extracted by a parameterized map

$$b_i(t) = \mathbf{b}_{\theta_5}(o_i(t)), \tag{8.7}$$

where $\theta_5 \in \mathbb{R}^{n_c}$ is a trainable vector. The nonlinear map $\mathbf{b}_{\theta_5} : \mathbb{R}^{f\times f} \to \mathbb{R}^n$ results from the following three layers: two single layer convolutional neural networks followed by vectorization and a fully connected layer.

Example 8.4.2. Fig. 8.1, illustrates a case in which the spatial state is simply the location of the agent (relative to the image) and observation map (8.6) crops a subset of the image based on its position. Moreover, the map describing the motion of the robots, according to (8.2), has resulted in horizontal and vertical translations of the agents across the image.

Example 8.4.3. Let us consider the settings of Example 8.4.1, where a camera is mounted on the robot. Then, map $\mathbf{o}(.)$ in (8.6) for this case is the projection map of the camera. For a camera, this map is completely characterized by the position and orientation of the robot with respect to the environment (i.e., camera extrinsics).

8.4.5 Structure of Information Inputs

In the previous subsections, we explained the details of the belief dynamics, agent motion and actuation, observation processing, communication, and decision-making modules on each agent. The same information input is fed to both LSTM modules in (8.1) and (8.4). We design the information input as a vector in $\mathbb{R}^{3n}$ with components

$$u_i(t) = \begin{bmatrix} b_i(t)^T & \bar{d}_i(t)^T & \lambda_i(t)^T \end{bmatrix}^T. \tag{8.8}$$

All these three components can be calculated using locally accessible data as we elaborate below. In Subsection 8.4.4, it was shown that $b_i(t)$ contains the features of the (partial) observation.

After communicating with neighbors, each agent decodes the received messages using a

parameterized map $\mathbf{d}_{\theta_6} : \mathbb{R}^{n_m} \to \mathbb{R}^n$ to get

$$d_i(t) = \mathbf{d}_{\theta_6}(m_i(t)), \tag{8.9}$$

where θ_6 is a trainable vector. We consider a fully connected layer with a ReLU activation for this map. Then, each agent takes the average of the received messages to find

$$\bar{d}_i(t) = \frac{1}{\Delta_i} \sum_{(j,i)\in\mathcal{E}} d_j(t), \tag{8.10}$$

in which Δ_i is the in-degree of node i in graph $\mathcal{G}$. This, $\bar{d}_i(t)$ is the aggregate message received by agent i at time t.

It is useful for agents to tag their beliefs and information by their spatial state. This can be done by the following map

$$\lambda_i(t) = \boldsymbol{\lambda}_{\theta_7}(p_i(t)) \tag{8.11}$$

where $\boldsymbol{\lambda}_{\theta_7} : \mathbb{R}^d \to \mathbb{R}^n$ is a parametrized map with a trainable vector θ_7.

In the final step, we close the loop by applying information input (8.8) to (8.1) and (8.4).

8.5 Decentralized Prediction and Classification

Recall that the image should be classified from M categories, while we have T rounds of observation and communication. To do this, first the raw prediction vector by agent i is evaluated using the final cell state and a map $\mathbf{q}_{\theta_8} : \mathbb{R}^n \to \mathbb{R}^M$ as

$$q_i = \mathbf{q}_{\theta_8}(c_i(T)). \tag{8.12}$$

We use a fully-connected linear layer with ReLU activation followed by a fully connected linear layer in place of this map. Then, we run a distributed average consensus algorithm over a strongly connected directed graph. Upon this averaging, on each agent, we will have

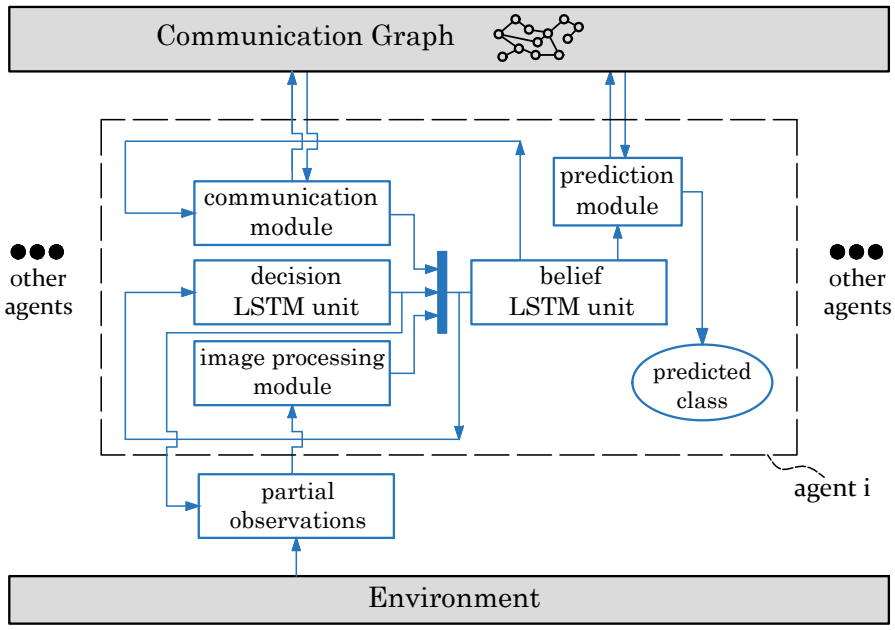

Figure 8.2: The diagram illustrating the essence of our framework.

a shared prediction vector $\bar{q} \in \mathbb{R}^M$, which is given by

$$\bar{q} = \frac{1}{N} \sum_{i=1}^{N} q_i. \tag{8.13}$$

It has been shown that if the communication graph is strongly connected, this task can be conducted in a completely decentralized manner [191]. Finally, each agent evaluates the system-wide prediction category using

$$q_c = \underset{j \in \{1,\ldots,M\}}{\operatorname{argmax}} \quad \operatorname{softmax}(\bar{q}). \tag{8.14}$$

One should note that in the current approach, we do not force the agents to reach a consensus on the predicted category, but rather we combine the beliefs due to sequences of partial observations to produce a single prediction.

In Fig. 8.2 we have illustrated the information flow of the framework that has been described throughout the section. These settings and steps can be summarized to build Algorithm 9, whose output is prediction category $q_c \in \{1, \ldots, M\}$ (shared by all agents).

8.6 Reinforcement Learning

We derive a generalization of the vanilla policy gradient algorithm, which utilizes the intrinsic differentiability of the rewards for simultaneous training of both prediction and motion planning parameters. First, let us stack our parameters as a single design parameter according to

$$\Theta := \left[\theta_1^T, \theta_2^T, \ldots, \theta_8^T\right]^T. \tag{8.15}$$

Next, let us denote all trajectories with positive probability of occurrence by $\mathcal{T}$. Suppose that in a sample execution $\tau \in \mathcal{T}$, image I corresponds to category $j \in \{1, \ldots, M\}$ (i.e., its actual category is j). Then, we define the reward corresponding to the outcome of this sample trajectory

$$r_\tau := -\mathbf{f}_l(\bar{q}_\tau - e_j), \tag{8.16}$$

where $\mathbf{f}_l$ is a differentiable nonnegative loss function (e.g. mean squared error (MSE)), $\bar{q}_\tau$ is the prediction at the end of this sampled trajectory, and $e_j \in \mathbb{R}^M$ is the unit coordinate vector in direction j. Based on the goal of this problem, we define our objective function as

$$J(\Theta) = \mathbb{E}\{r_\tau\} = \sum_{\tau \in \mathcal{T}} p_\tau \, r_\tau. \tag{8.17}$$

Therefore, we need to solve the optimization problem

$$\underset{\Theta}{\text{maximize}} \;\; J(\Theta). \tag{8.18}$$

The gradient of J with respect to Θ can be written as

$$\nabla_\Theta J \; = \sum_{\tau \in \mathcal{T}} r_\tau \nabla_\Theta p_\tau + p_\tau \nabla_\Theta r_\tau. \tag{8.19}$$

Let us drop index Θ for simplicity. Using the well-known gradient derivation technique

similar to that of the REINFORCE algorithm [192], we can write

$$\begin{aligned} \nabla J &= \sum_{\tau \in \mathcal{T}} p_\tau \nabla (\log p_\tau) r_\tau + p_\tau \nabla r_\tau \\ &= \mathbb{E}\{\nabla (\log p_\tau) r_\tau + \nabla r_\tau\}. \end{aligned} \tag{8.20}$$

Let us execute Algorithm 9 for N_r independent experiments. Then, for each sample $k = 1, \ldots, N_r$, we use $p^{(k)}$ to denote the probability that this particular trajectory is selected. Now, inspired by (8.20), we define the proxy sampler for J to be

$$\hat{J} := \frac{1}{N_r} \left(\sum_{k=1}^{N_r} \log p^{(k)} r_d^{(k)} + r^{(k)} \right), \tag{8.21}$$

where quantity $r_d^{(k)}$ has a value equal to $r^{(k)}$, but has been detached from the gradients. This means that a machine learning framework should treat $r_d^{(k)}$ as non-differentiable during training. Then, we inspect that

$$\mathbb{E}\left\{\nabla \hat{J}\right\} = \nabla J, \tag{8.22}$$

i.e., $\nabla \hat{J}$ is an unbiased estimator of $\nabla (\log p_\tau) r_\tau + \nabla r_\tau$ that appears in (8.20). Therefore, it is justified to follow the approximation for the gradient given by

$$\nabla J \approx \nabla \hat{J}. \tag{8.23}$$

Note that the first term in summation (8.21) is identical to the quantity that is derived in the policy gradient method with a reward that is independent of the parameters (i.e., identical to REINFORCE algorithm). The second term accounts for the fact that the reward in our settings directly depends on parameter Θ: for two different set of parameters, if the agents receive exactly the same sequences of observations and take exactly the same actions, still the reward explicitly depends on the parameters of the network (e.g. weights of the convolution layers or fully connected layers).

8.7 Numerical Experiments

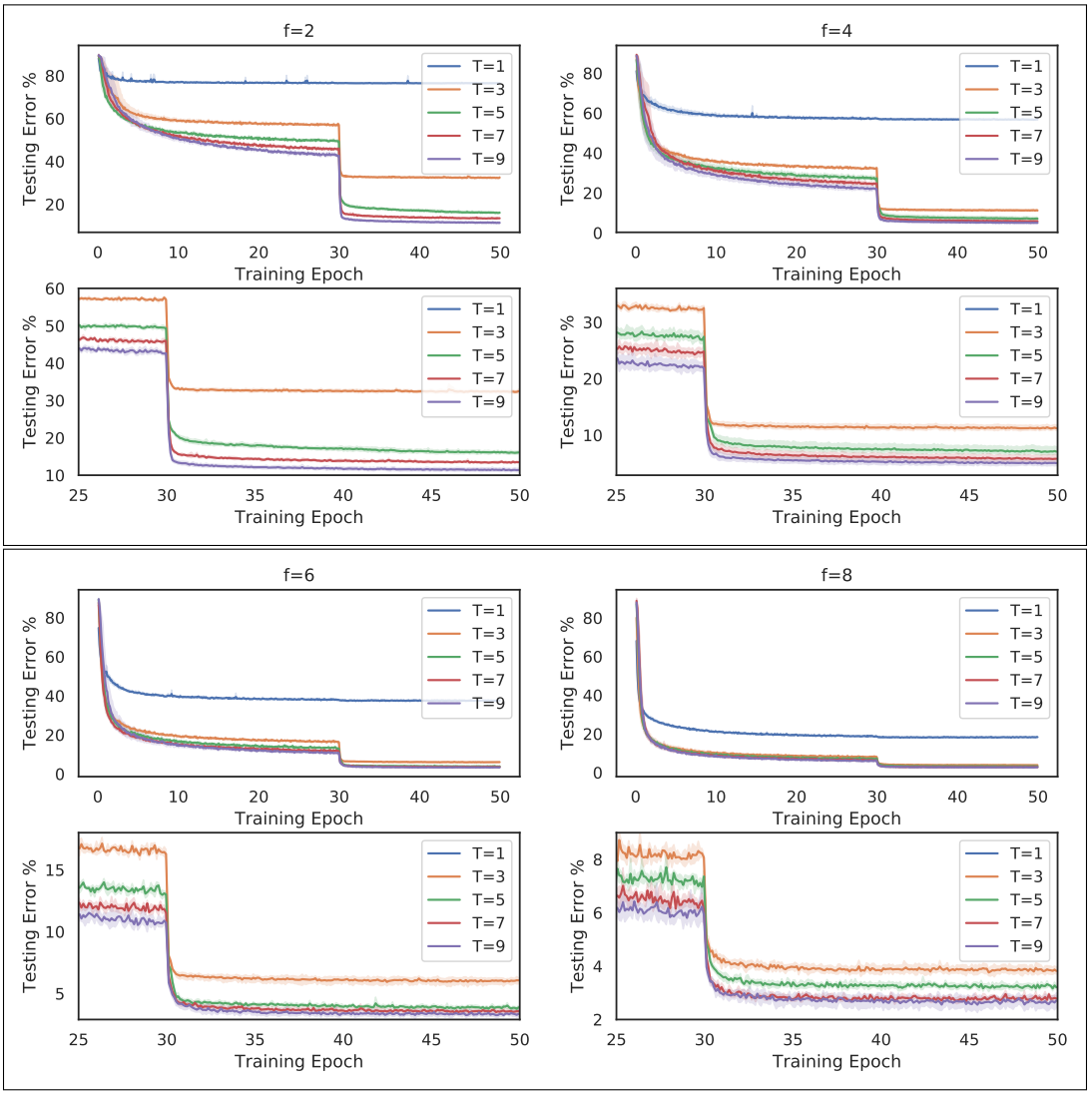

Figure 8.3: The testing accuracy for different frame sizes f and time horizons T versus the number of training epochs.

We use the MNIST dataset of handwritten digits [193] to test the proposed learning algorithm. The dataset consists of 60,000 training images and 10,000 testing images, where each image has 28×28 pixels. We suppose that each agent may observe a portion of the image that has $f \times f$ pixels. The spatial variable p_i is the pixel coordinate of the top left corner of the observation window. The possible movements by each agent can be

characterized by

$$\mathcal{A} = \{\text{up}, \text{down}, \text{left}, \text{right}\}. \tag{8.24}$$

By each movement, the agent is translated across the desired direction by f_m pixels. If the sampled action is infeasible, then the agent remains at its location at that time instant. In Fig. 8.1, as an example, we have illustrated an image from these data and the observations that three agents receive during the horizon. In all experiments of this section, we choose a batch-size of 64 images during training. We also choose the variable size of LSTM unit to be $n = 64$, the number of neurons of all fully connected layers to be 64, and the dimension of the broadcasted messages to be $n_m = 12$. We have implemented this approach using the machine learning framework PyTorch [194].

Testing Accuracy Results: We consider $N = 2$ agents that are communicating over the only option for a strongly connected graph; i.e., graph with arc set $\mathcal{E} = \{(1, 2), (2, 1)\}$. We choose $N_r = 30$ and conduct a parametric study by varying the observation frame size f and time horizon T according to $f \in \{2, 4, \ldots, 8\}$ and $T \in \{1, 3, \ldots, 9\}$, respectively. For each pair of f and T, we train the model for 50 epochs. However, we break down the training into two stages: first, we consider random walks for 30 epochs. Then, we fix all of the parameters except for the decision-making module and train the model for another 20 epochs. In Fig. 8.3, we demonstrate the progress of the testing error versus the number of training epochs. Also, in Fig. 8.4, we show the average maximum testing accuracy for each pair of frame size f and time horizon T in the case of random walks (i.e, at the end of 30 epochs) and also with the designed law for the movements of the agents (i.e., after additional 20 epochs of training for motion planning). The results suggest that having a policy that governs the movements of the agents may significantly decrease the testing error; e.g. Fig. 5 implies that for $f = 4$ (i.e., 16 observation pixels) and $T = 7$ communication and observation steps, the testing error has decreased by more than 70%.

Remark 24. The intuition behind our two-stage method of training is that we initially train the perception, communication, and prediction modules while agents are learning to explore the environment. Once these modules are sufficiently trained, we let the agents learn how to

traverse the image. The numerical experiments suggest that this training method generally results in smaller testing errors, compared to the case in which we simultaneously optimize the parameters of all modules. One possible justification this this observation is that the two-stage training method is less prone to getting stuck in local non-stationary solutions due to higher exploration in the first phase.

Alternative Classification Schemes: We consider two alternative methods to classify the images based on similar observations, which will be compared against our approach. In both cases, each agent independently conducts a *random walk*. *(i) Centralized Classification with Random Walks:* An alternative is to collect all the images from all agents based on random movements that have equal probability in each of four directions (i.e. 1/4). Then, we feed the resulting unmasked image to a single CNN which is embedded into a prediction vector $q \in \mathbb{R}^M$ (similar to $\bar{q}$ in (8.13)). In Fig. 8.5, we demonstrate a typical input of this simple centralized classification, which has the same dimensions as the images in MNIST (i.e., $f = 28$). *(ii) Distributed Classification with Random Walks:* We also consider a variant of Algorithm 9 in which instead of learning the optimal distributions for the policy, we suppose that all possible movements in (8.24) have an equal probability of 1/4. We set the parameters to $f = 2$, $N = 2$ agents, $N_r = 10$ samples, time horizon $T = 4$, and a complete communication graph. We conduct the training with random walks for 20 epochs, which is followed by training of the decision-making motion planning module for another 20 epochs. We train the centralized classification for 40 epochs as well. In Fig. 8.6, we illustrate the results, which show that the quality of prediction using the proposed method is superior relative to the method with random walks as well as the centralized method.

Remark 25. The main reason for which the performance of the centralized method is not better than our approach is that the masks created by random motion of the agents are not optimal.

Effect of Communication: We compare the result of training for an alternative structure in which the agents do not communicate during the horizon (although, they finally do so conduct the prediction). We set the parameters to be $N = 2$, $f = 4$, $N_r = 40$, and $T = 6$. As shown in Fig. 8.7, it turns out that smaller testing errors compared to the case that the

agents do not communicate.

Effect of Number of Agents: We consider the set of parameters $f = 4$, $N_r = 25$, and $T = 4$ and conduct the training for a different number of agents communicating over a complete directed graph. In Fig. 8.8, we illustrate the result of training for 30 epochs with random walks followed by learning the moving policies for 20 epochs. As expected, we observe that increasing the number of agents significantly reduces the testing error.

Visualization of Communicated Messages: We explore the patterns in the messages that are broadcasted by the agents using the learned communication medium. The goal is to visualize how the agents express the shared memory within an episode for solving the task. After training, we simulated 500 sample trajectories with $T = 9$ and recorded the messages of all agents at every $t = 0, \ldots, 8$. Then, we used the dimensionality reduction technique called t-SNE [195] to produce meaningful visualizations of the messages. In Fig. 8.9, we illustrate two t-SNE plots for the messages at $t = 0$ and $t = 6$, which correspond to the messages before any communication and after a few rounds of communication and observation, respectively. In these figures, every message, which is initially in $\mathbb{R}^{12}$, is reduced to a vector in $\mathbb{R}^2$ and is illustrated with a color corresponding to the true label of the image. The result of clustering at $t = 0$ implies that initially, there is no meaningful pattern in the distribution of the labels. However, as the agents move across the image and communicate, they construct an internal belief about the true category. The second figure shows a t-SNE plot after 6 time-steps, which suggests that agents' beliefs are reflected in the communicated messages. In fact, we observe that the digits are now clearly clustered in their own groups; i.e., they have learned to broadcast their beliefs about the true labels to their neighboring agents.

8.8 Concluding Remarks

We introduce and analyze a multi-agent image classification framework using a generalized policy gradient as the core reinforcement learning technique. The underlying ideas that are discussed in this chapter are applicable to the other value-based, policy-based, or while using novel variance reduction techniques [196]. The problem studied in this chapter has

a discrete action space within a typically short time horizon. Depending on the problem structure (e.g. in aerial robotic applications), one may prefer a continuous action space. Then, it is straightforward to generalize our methodology to deal with these policies within this framework; for instance, using Gaussian policies [197].

Our extensive simulations suggest that the current models are temporally robust: if we train the model for a time horizon $T = T_1$ and execute the model for $T = T_2 > T_1$, the prediction quality using the second model will remain at a meaningfully high level. Also, changing the number of agents will not result in dramatic performance degradation. For example, a model trained with 3 agents will still produce acceptable outcomes for problems with 2 and 4 agents. Due to space limitations, we have not included the related numerical experiments.

In this chapter, our numerical experiments have been limited to 2-D image classification. However, the proposed framework can be applied, with minor adjustments, to more realistic scenarios. For instance, as explained in Example 8.4.3, the current framework allows the classification of 3-D objects using a sequence of intelligently chosen 2-D observations conducted by moving agents. The optimal (stochastic) movement of the agents around the object could be potentially related to the *next best view* problem [198]. Therefore, we expect that the sample efficiency of our methodology can be potentially enhanced by incorporating the developed optimal movement theories (e.g. see [199]). It is also an interesting line of research to study the cases where the graph structure dynamically (e.g. randomly) evolves over time.

Algorithm 9 Multi-Agent Classification (Execution)

input: Input image $I \in \mathbb{R}^{n_I \times n_I}$
initial spatial states $p_1(0), \ldots, p_N(0)$
output: prediction category q_c

initialize:
for $i \in \mathcal{V}$ **do**
initialize the states $h_i(t) \leftarrow 0$, $c_i(t) \leftarrow 0$
for $j \in \mathcal{N}_i$ **do**
initialize the messages $m_j(0) \leftarrow 0$
end for
end for
for $t = 0$ to $T - 1$ **do** ▷ communication & observation
for $i \in \mathcal{V}$ **do**
conduct the observation $o_i(t) \leftarrow \mathbf{o}(I, p_i(t))$
map the observation $b_i(t) \leftarrow \mathbf{b}_{\theta_5}(o_i(t))$
for $j \in \mathcal{N}_i$ **do**
decode message $d_j(t) \leftarrow \mathbf{d}_{\theta_6}(m_j(t))$.
end for
find average message $\bar{d}_i(t) \leftarrow \dfrac{1}{\Delta_i} \displaystyle\sum_{(j,i)\in\mathcal{E}} d_j(t)$
map the spatial state $\lambda_i(t) \leftarrow \boldsymbol{\lambda}_{\theta_7}(p_i(t))$
form input $u_i(t) \leftarrow \begin{bmatrix} b_i(t)^T & \bar{d}_i(t)^T & \lambda_i(t)^T \end{bmatrix}^T$.
run the belief LSTM unit (8.1)
evaluate message $m_i(t+1) \leftarrow \mathbf{m}_{\theta_4}(h_i(t+1))$
run the decision LSTM unit (8.4)
update policy distribution $\boldsymbol{\pi}_{\theta_3}(. \,|\hat{h}_i(t+1))$
samples action $a_i(t+1)$ based on π
update spatial state $p_i(t+1) \leftarrow \mathbf{g}(p_i(t), a_i(t+1))$
end for
end for
for $i \in \mathcal{V}$ **do** ▷ local raw predictions
find raw prediction vector $q_i \leftarrow \mathbf{q}_{\theta_8}(c_i(T))$
end for
conduct the distributed average consensus $\bar{q} \leftarrow \dfrac{1}{N} \displaystyle\sum_{i=1}^{N} q_i$
find the prediction category

$$q_c \leftarrow \text{softmax}\big(\underset{i\in\{1,\ldots,M\}}{\text{argmax}} \; \bar{q}\big)$$

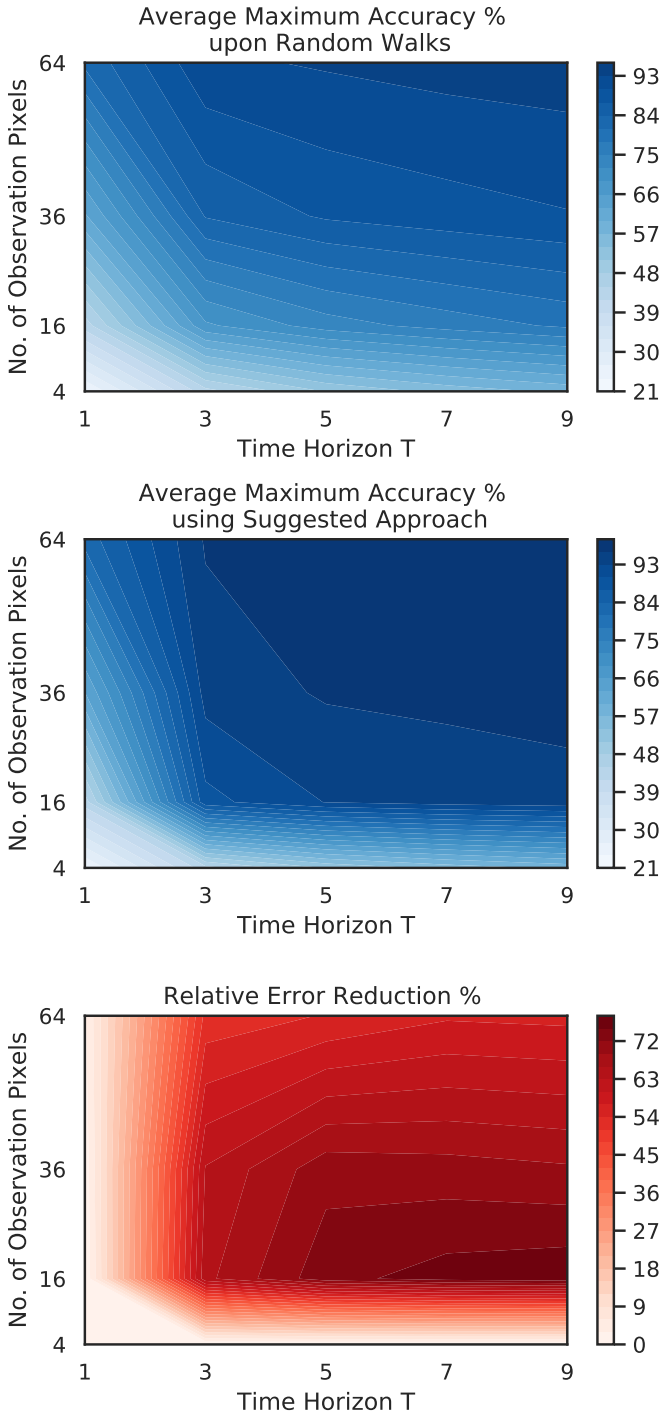

Figure 8.4: Average maximum testing accuracy versus frame size f and time horizon T after 30 epochs with random walks (top figure). In the middle one, we trained extra 20 training epochs for the motion planning policy. The last figure illustrates the error reduction as a result of the design of coordination policy instead of random walks.

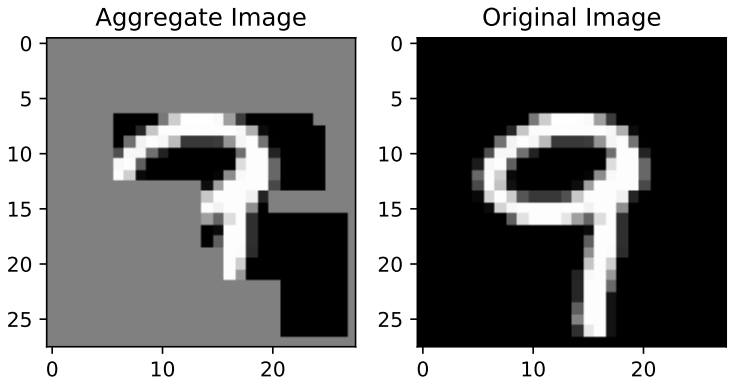

Figure 8.5: A sample of masked images created by putting together the observations by 3 different agents for a time horizon of $T = 3$ with $f = 8$. This image is the input to the centralized image classifier as an alternative classification approach (method *(i)*). The (random) uncovered parts have been reached due to random walks.

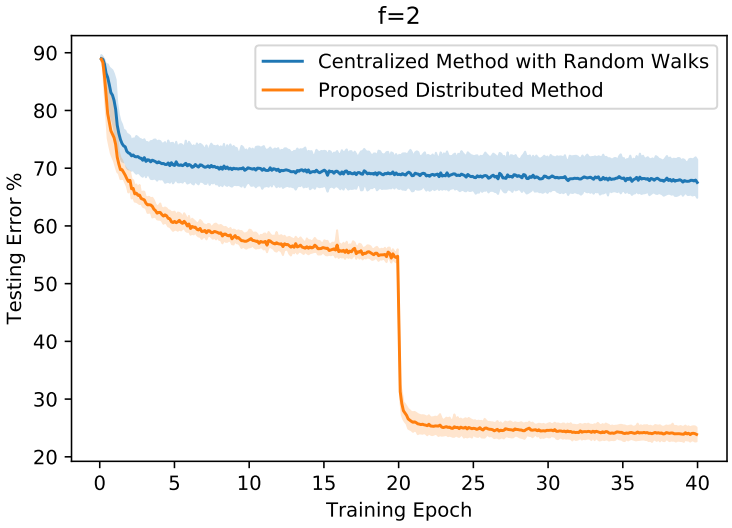

Figure 8.6: Comparison of the testing error when using a centralized method with the suggested approach. The first 20 epochs of training corresponds to the case for random walks, while in the next 20 epochs we optimize the movements of the agents.

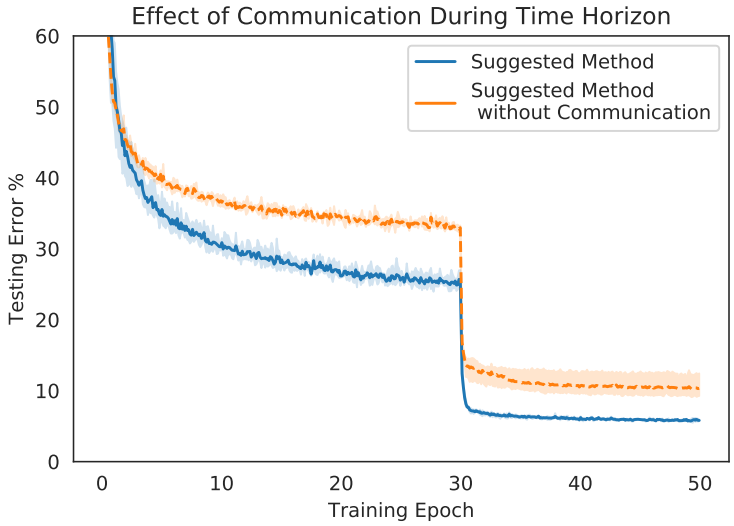

Figure 8.7: The result of training when with and without communications.

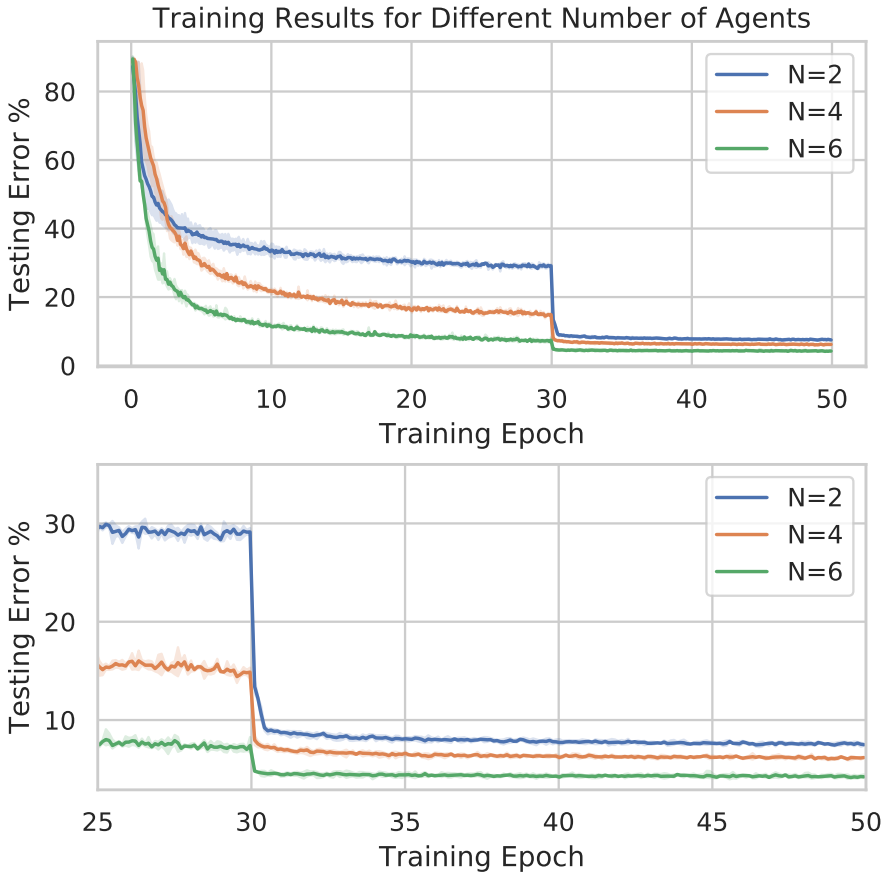

Figure 8.8: The training results for different number of agents.

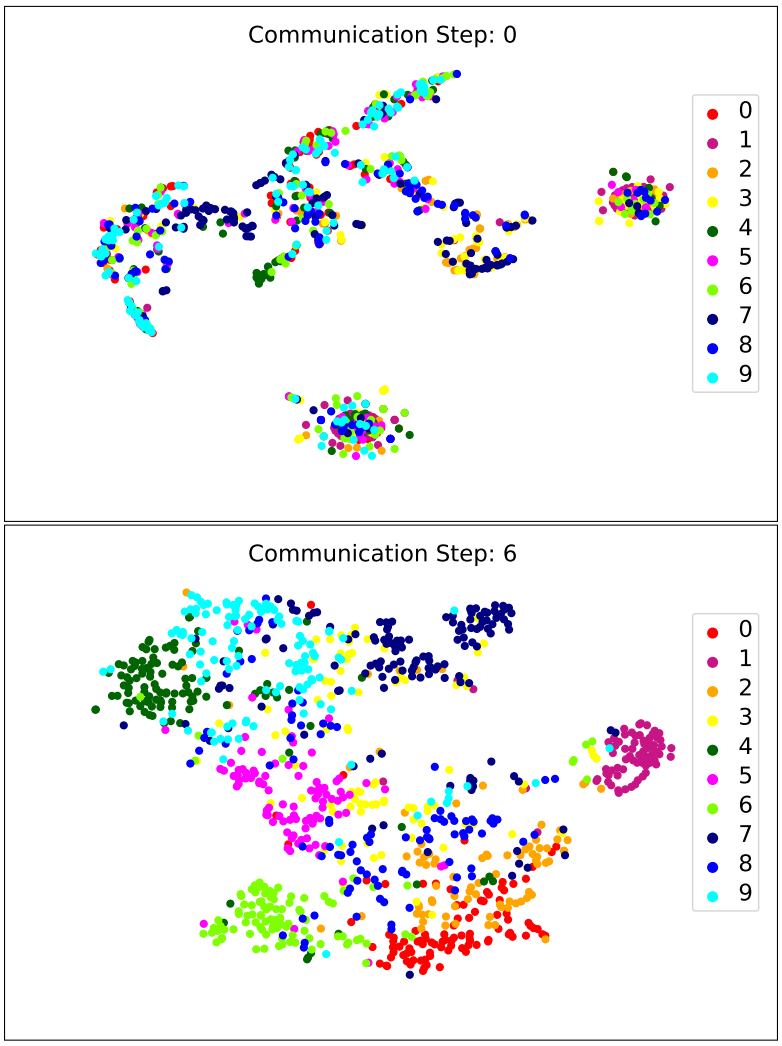

Figure 8.9: Visualization of the learned communications strategies and how they share their beliefs with each other using t-SNE plots.

Chapter 9

A Layered Architecture for Active Perception: Image Classification using Deep Reinforcement Learning

There has been a rapidly growing interest in goal reasoning in recent years; planning mechanisms for agents that are capable of explicitly reasoning about their goals and changing them whenever it becomes necessary [200,201]. The potential applications of goal reasoning spans over several research fields, for example, only to name a few, controlling underwater unmanned vehicles [202], playing digital games [203], and air combat simulations [204].

One of the promising recent frameworks for goal-based planning and reasoning is hierarchical deep Q-networks (hDQN) [205], which consists of two layers: a meta-layer that plans strategically and an action-layer that plans local navigation. The meta-layer receives a state as its input and outputs a goal, a condition that can be evaluated in a given state. The action-layer receives a state and a goal as its input. Then, it selects and executes actions until the agent reaches a state where the goal is achieved. Both layers use a deep neural network similar to that of DQN with some important differences: the meta-layer selects goals in order to maximize external rewards from the environment, while the action-layer

selects actions to maximize designer-defined intrinsic rewards (e.g., 1 for reaching the goal state and 0 otherwise).

In this work, we consider the problem of exploring an environment by a robot for classification purposes. Contrary to the standard assumptions made in the literature, we assume that robot can only partially observe the environment, where each observation depends on the actions taken by the robot. The first and second layers of our proposed architecture are similar to those of hDQN, while the third layer perform a classification task and evaluates the reward in a differentiable manner.

Our approach has other differences from hDQN. First, note that in hDQN requirement, the action-layer reaches a state achieving the goal. However, find that this assumption is too restrictive, unnecessary, and potentially unrealizable due to partial observability for our purposes. Instead, our method relaxes this requirement by allowing a robot to move a few steps towards the goal, but not necessarily reaching to it. This flexibility is needed because our intrinsic objective is to explore the environment. Therefore, the goal planner should only dictate a desired general direction of exploration rather than imposing a hard constraint to reach a specific position. In this sense, our goals play a similar role to tasks in hierarchical task network planning [206], where the tasks are processes inferred from the agent's execution (e.g., "explore in this direction") rather than goals, which need to be validated in a particular state (e.g., "reach coordinate $(3, 5)$"). Second, the nature of our problem motivates a single unified reward for the meta-layer and action-layer rather than separate rewards. As already mentioned, this reward is the output of the classification layer. Lastly, the partial observability of our problem motivates derivation and use of policy-gradient approaches for learning the model parameters. As illustrated in [207], such generalized policy gradient algorithms allow co-design of goal generator, action planner, and classifier modules.

Our methodology incorporates goal reasoning capabilities with deep reinforcement learning procedures for robot navigation by introducing intermediate goals, instead of requiring the robot to take a sequence of actions. In this way, our architecture provides transparency in terms of what the robot is trying to accomplish and, thereby, provides an explanation for its own course of action. The statement of the classification problem is identical to that

of [207], but with some important differences. In [207], we employ multiple agents with a recurrent network architecture, while robots do not enjoy goal reasoning capabilities.

Related Literature: We cast the classification problem as a planning and perception mechanism with a three-layer architecture that is realized through a feedback loop. We are particularly interested in planning for perception. A related line of research is active perception: how to design a control architecture that learns to complete a task and at the same time, to focus its attention to collect necessary observations from the environment (see [208, 209] and references therein). The coupling between action and perception has been also inspired by human body functionalities [210].

Visual attention is another related line of work. It is based on the idea that for a given task, in general, only a subset of the environment may have necessary information, motivating the design of an attention mechanism [211,212]. These have been motivating for saliency-based techniques for computer vision and machine learning, where the non-relevant parts of the data are purposely ignored [213–217].

Notations: The i'th element of a vector π is denoted by $\pi[i]$, where indexing may start from 0. For an integer $T > 0$, $[T]$ denotes the sequence of labels $[0, 1, \ldots, T-1]$. For two images $y_1 \in \mathbb{R}^{c_1 \times n \times n}$ and $y_2 \in \mathbb{R}^{c_2 \times n \times n}$ that have the same dimensions but different number of channels, their concatenation is denoted by $\text{concat}([y_1, y_2]) \in \mathbb{R}^{(c_1+c_2)\times n \times n}$. The categorical distribution over the elements of a probability matrix (or vector) π, whose elements add up to 1, is denoted by $\text{categorical}(\pi)$. For two probability vectors, $\pi_1, \pi_2 \in \mathbb{R}^D$, the cross-entropy between the corresponding categorical distributions is denoted by $\text{CrossEntropy}(\pi_1, \pi_2)$.

9.1 Problem Statement

Let us consider an agent (robot) that is capable of moving in some pre-specified directions (such as up, down, right, and left) in order to explore an image (e.g., map of a region) during a sequence of $E > 0$ episodes, where the duration of each episode is $T > 0$ steps in time. For integers $c, n > 0$, we represent an instance of an image by a $c \times n \times n$ matrix. Suppose that at the beginning of episode $e \in [E]$ a goal $g(e)$ is assigned to the robot and at every

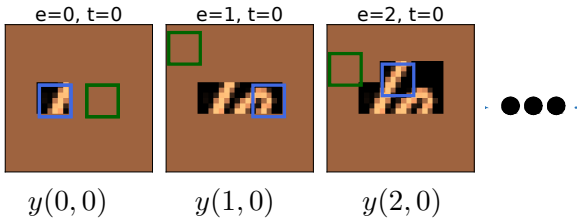

Figure 9.1: Snapshots of the proposed problem at the beginning of three episodes. The blue and green squares point to the current position of the agent and the goal of each episode. During each episode, the agent has moved towards the goal.

time step $t \in [T]$ (within that episode), the robot moves towards $g(e)$ to discover a portion of image $x \in \mathbb{R}^{c \times n \times n}$ based on its current pose $p(e,t) \in \mathbb{R}^2$. The robot takes an action to update its position. Based on its past history, the agent has uncovered portions of x up to time t, which is denoted by $y(e,t) \in \mathbb{R}^{c \times n \times n}$. The undiscovered portions of x in $y(e,t)$ are set to 0. Fig. 9.1 illustrates this scenario through an example, where the discovered image $y(e,t)$, the robot's position, and its goal are demonstrated at different episodes and times.

The *problem* is to design a layered architecture that generates meaningful goals and plans navigation towards assigned goals, with the objective of performing image classification.

9.2 A Multi-layered Architecture

We propose an architecture where a robot collects local observations from an image, generates intermediate goals based on what it has been observed, takes local actions to move towards these goals, and, finally, makes a prediction based on the discovered information by the end of the last episode to classify the underlying image. This architecture consists of three layers, where each receives a different set of information as their inputs. These inputs are defined using some auxiliary internal variables. For given $e \in [E]$ and $t \in [T]$, we define an auxiliary image $l(e,t) \in \mathbb{R}^{n \times n}$ whose pixels are set to 1 everywhere except over a $m \times m$ patch of pixels with 0 values, where m denotes the width and height of the partial observation by the agent This variable solely depends on the robot position $p(e,t)$. Similarly, we define an auxiliary image $h(e,t) \in \mathbb{R}^{n \times n}$ where the value of a pixel is set to 0 if robot has visited that pixel before, otherwise to 1. This variable keeps track of the history of the agent.

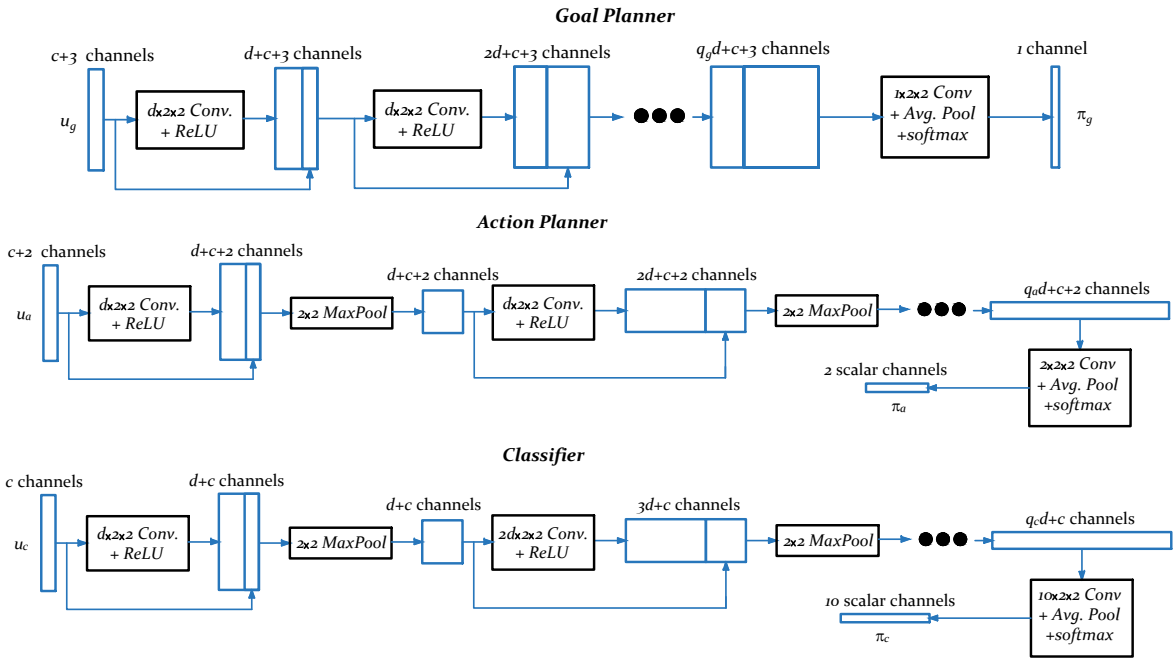

Figure 9.2: A schematic diagram of the 3-layered deep learning architecture for goal generator, action planner, and classifier. The dots correspond to repeating the preceding modules for r times. In the planners, the number of channels in the convolutional filters is fixed and equal to d in the consecutive layers. For the classification module, the number of output channels from the convolutions is doubled each time. Thus, in each case we will have different numbers of intermediate channels q_g, q_a, and q_c (the components are not drawn).

9.2.1 Goal Planner

We consider a fully-convolutional architecture of ResNet style [218] for the planner, where the skip connections are modified to have concatenation form instead of summation (similar to densely connected architecture [219]). The top portion of Fig. 9.2 illustrates our architecture. At the beginning of episode e, information input $u_g(e) \in \mathbb{R}^{(c+2)\times n\times n}$ is formed by concatenating three inputs:

(i) Undiscovered image up to this episode and instant, which is defined by

$$y(e-1) := y(e-1, T-1) \in \mathbb{R}^{c\times n\times n}. \tag{9.1}$$

We recap that $y(e,t)$ is the undiscovered portions of the underlying image at episode e and time t.

(ii) An image that encapsulates the position of the robot in the environment by the end of the previous episode, which is defined by

$$l(e-1) := l(e-1, T-1) \in \mathbb{R}^{n\times n}. \tag{9.2}$$

(iii) An image that encapsulates the history of all visited positions up to that episode, which is defined by

$$h(e-1) := h(e-1, T-1) \in \mathbb{R}^{n\times n}. \tag{9.3}$$

We feed the following input to the planner

$$u_g(e) := \text{concat}(\,[\,y(e-1), l(e-1), h(e-1), g_l(e-1)\,]\,),$$

where $g_l(e-1)$ is derived from the previous goal $g(e-1)$ according to a procedure that is explained at the end of this subsection. Then, we utilize the convolution architecture that outputs a single channel $n \times n$ image. By applying softmax on this image, we arrive at an

$n \times n$ probability matrix that can be characterized by a nonlinear map

$$\pi_g(e) = f_1\big(u_g(e); \theta_1\big), \tag{9.4}$$

where θ_1 is a trainable parameter. We define a categorical probability distribution over the pixels using $\pi_g(e)$, which will allow us to sample goal $g(e) \in \mathbb{R}^2$ from this distribution

$$g(e) \sim \text{categorical}\big(\pi_g(e)\big). \tag{9.5}$$

As a feedback signal for this layer and action-layer in the next episode, auxiliary variable $g_l(e) \in \mathbb{R}^{n \times n}$ is created, which is an image whose pixel values are set to 0 only at the $m \times m$ patch corresponding to the goal $g(e)$ and 1 elsewhere (similar to $l(e)$).

9.2.2 Action Planner for Local Navigation

During each episode, the robot takes T actions towards an assigned goal. It is assumed that the actions taken by the robot are at most a fixed number of pixels to the left, right, up, or down. Given the goal of the episode, one can inspect that there is always at most one horizontal action (either left or right) and one vertical action (either up or down) that we count as moving towards the goal. Therefore, given current position $p(e,t)$ and goal $g(e)$, the problem of planning local actions can be formulated as finding a probability vector $\pi_a(e,t) \in \mathbb{R}^2$ that will allow the robot to choose between vertical and horizontal actions and move towards the goal. In situations where only one of these actions takes the robot closer to the goal, we do not use this distribution. More precisely, robot's action protocol is given by

$$a(e,t) = \begin{cases} \text{vertical action} & \text{if } p(e,t)[0] = g(e)[0] \\ \text{horizontal action} & \text{else if } p(e,t)[1] = g(e)[1] \\ \text{sample from dist.} & \text{otherwise} \end{cases}.$$

To evaluate the probability vector π_a, we consider a similar fully-convolutional architecture for choosing the local actions; we refer to the middle portion of Fig. 9.2. The input to this

architecture lives in $\mathbb{R}^{(c+3)\times n\times n}$ and is defined by

$$u_a(e,t) = \text{concat}\big(\,[\,y(e,t),\, l(e,t),\, h(e,t),\, g_l(e)\,]\,\big). \tag{9.6}$$

The convolutional mapping results in an image with 2 channels. Then, we use global average-pooling from this output, which is followed by softmax normalization to get a vector $\pi_a(e,t) \in \mathbb{R}^2$. By composing all these maps, we can obtain the following characterization

$$\pi_a(e,t) = f_2\big(u_a(e,t);\, \theta_2\big), \tag{9.7}$$

where θ_2 is a trainable parameter. We construct a categorical distribution which will enable the robot to select among vertical or horizontal actions via random sampling, i.e.,

$$a(e,t) \;\sim\; \text{categorical}\big(\pi_a(e,t)\big). \tag{9.8}$$

9.2.3 Image Classifier

A similar convolutional architecture is considered for the classification module; we refer to the bottom portion of Fig. 9.2. Classification is conducted at the end of the last episode, i.e., at episode $E-1$ and time step $T-1$. Let us tag the last explored image by $y_f := y(E-1, T-1) \in \mathbb{R}^{c\times n\times n}$. This will be the input to the classifier, i.e.,

$$u_c = y_f. \tag{9.9}$$

The output of the convolutional layer has D channels, which is global average pooled before applying softmax to get the prediction vector $\pi_c \in \mathbb{R}^D$. Similar to the other two layers, the corresponding nonlinear map can be represented by

$$\pi_c = f_3\big(u_c;\, \theta_3\big), \tag{9.10}$$

where θ_3 is a trainable parameter. The reward is defined as

$$r = -\text{CrossEntropy}\Big(\pi_c\,,\,\pi_c^l\Big), \tag{9.11}$$

in which $\pi_c^l \in \mathbb{R}^D$ is the label probability vector. This vector is equal to unit coordinate vector in j'th direction, where $j \in [D]$ is the label.

9.3 Reinforcement Learning Algorithm

Similar to previous chapter, we use a learning algorithm to train various layers in our architecture. The objective is to find an unbiased estimator for the expected reward whenever the reward of the reinforcement learning explicitly depends on the parameters of the neural network. Let us put all trainable parameters in one vector and represent it by $\Theta := \big[\theta_1^T, \theta_2^T, \theta_3^T\big]^T$. The set of all trajectories is shown by $\mathcal{T}$ and the corresponding reward to a given trajectory $\tau \in \mathcal{T}$ by r^τ. The objective is to maximize the expected reward, i.e.,

$$\underset{\Theta}{\text{maximize}}\; J(\Theta),$$

where $J(\Theta) = \mathbb{E}\{r^\tau\} = \sum_{\tau\in\mathcal{T}} \pi^\tau\, r^\tau$ and π^τ is the probability of choosing goals and actions given the value of the current parameter Θ. Suppose that N independent trajectories are created, i.e., N rollouts[1], where $\pi^{(k)}$ and $r^{(k)}$ denote the probability of this particular trajectory and the resulting reward, respectively, for $k = 1, \ldots, N$. Let us define $\hat{J}$ to be

$$\hat{J} \;:=\; \frac{1}{N}\Big(\sum_{k=1}^{N} \log \pi^{(k)} r_d^{(k)} + r^{(k)}\Big), \tag{9.12}$$

where the value of the quantity $r_d^{(k)}$ is $r^{(k)}$, while it has been detached from the gradients. This means that a machine learning algorithm should treat $r_d^{(k)}$ as a non-differentiable scalar

[1]A rollout is executing a fixed policy given an identical initial setting with a random seed. Different rollouts are required when the outcome of the game is uncertain (i.e., stochastic) [220].

Layer	being trained	trained & fixed	i.i.d.
Classifier	✓	✓	×
Goal Planner	✓	✓	✓
Action Planner	✓	✓	✓

Table 9.1: Different possibilities for training of different layers.

during training[2]. Then, in the previous chapter we showed that

$$\mathbb{E}\left\{\nabla \hat{J}\right\} = \nabla J, \tag{9.13}$$

i.e., $\nabla \hat{J}$ is an unbiased estimator of ∇J. This justifies the use of approximation $\nabla J \approx \nabla \hat{J}$.

Remark 26. The first term inside the summation in (9.12) is identical to the quantity that is derived in the policy gradient method with a reward which is independent of the parameters, i.e., the REINFORCE algorithm [192]. The second term indicates that reward directly depends on Θ. For example, if all goals and actions have equal probability of being selected, then it will suffice to consider only the second term inside the summation in (9.12).

9.3.1 Hierarchical Training

The proposed multi-layered architecture as well as this policy gradient algorithm allow us conduct training of the three layers (i.e, goal planner, action planner, and classifier) with a wide range of flexibility. All three modules can be either in training mode or kept fixed after training. Moreover, for goal and action planning layers, we have an extra level of flexibility *before* training: we can consider i.i.d. (i.e., independent and identically distributed) planning of goals or actions. This mode of operation for goal planner implies that the goals are chosen from a uniform distribution over all pixels. This model of operation for action planner means that taking horizontal or vertical actions towards the goal have always equal probability of 1/2. Once we switch to learning the parameters for either of these planners, we cannot switch back to i.i.d. mode. In Table 9.1, we have summarized these possibilities. In this paper, we consider a sequence of three different training modes:

(i) meta-layer and action-layer in i.i.d. mode, while classifier is being trained, (ii) action-

[2]The reason for this treatment is because of the idea behind the chain rule: in $(fg)' = f'g + g'f$, f and g in the right hand side correspond to being kept constant while the other term varies.

layer in i.i.d. mode, while classifier and goal planner are being trained simultaneously, (iii) all layers being trained simultaneously.

In every mode, reward r^τ is equal to r given by (9.11). In mode (i), all goals and actions are identically distributed. Thus, we can arbitrarily set $\log p^\tau = 0$ (or any other constant). In mode (ii), only the goals are actively decided. Therefore, the probability term is given by

$$\log \pi^\tau = \sum_{e \in [E]} \log \pi_g(e),$$

while for mode (iii), we need to set

$$\log \pi^\tau = \sum_{e \in [E]} \log \pi_g(e) + \sum_{e \in [E]} \sum_{t \in [T]} \chi(e,t) \log \pi_a(e,t),$$

where $\chi(e,t) = 1$ if the action at instant (e,t) was decided by action distribution, and $\chi(e,t) = 0$ otherwise.

9.4 Numerical Experiment

We test the method on the MNIST dataset of handwritten digits [193]. The dataset consists of $60,000$ training examples and $10,000$ test images, each of 28×28 pixels.

General Setup: The dataset was normalized between -0.5 and 0.5. In all experiments, the agent starts at a random position inside the image. The actions in any direction are 2 pixels per step. In these experiments, we did not use the test set for hyper-parameter tuning. We used student's t-test for the confidence interval of stochastic accuracies with α-value of 5%. The number of rollout per data point was 4 in the experiments (unless otherwise). We used Adam solver for the optimization with a mini-batch size of 60 images. The model was built in PyTorch [194].

Sample Accuracy Results: We conduct the training with patch size $m = 6$ for $E = 4$ episodes that each have a horizon of $T = 5$. The training and testing accuracies for the trained model where $94.39 \pm 0.03\%$ and $94.61 \pm 0.17\%$, respectively. This suggests an acceptable level of generalization for our trained model to unseen test set, while the accuracy on the test set has a slightly higher variance.

Sample Trajectories: In Fig. 9.3, we demonstrate 3 sample trajectories on 2 test data points next to resulting prediction probabilities. We have intentionally illustrated both high confidence and low confidence outcomes. For instance, on the test point with label 4, the second trajectory results in a wrong prediction, which is likely due to the fact that the agent has not uncovered the upper region of 4 in its limited temporal budget. As one observes, in most cases, the goals and actions are selected such that the agent can see the most informative parts of the image.

Top Two Category Accuracy: For the previously described model, we evaluate the top 2 class accuracy (i.e., if the true label is among the top 2 categories predicted by the model). Then, the training and testing accuracies increased to $98.27 \pm 0.02\%$ and $98.30 \pm 0.04\%$, respectively.

Confusion Matrix: For the trained model, we build the confusion matrix of the classification for the testing data. In Fig. 9.4, we show this matrix. The reported accuracies are averaged over 20 independent experiments.

Performance of Classifier Module: Let us consider the trained classifier module with complete (i.e, unmasked) image as its input; i.e., $u_c = x$. We can evaluate the performance of this isolated model, which turns out that the training and testing accuracies were 94.85% and 94.92%, respectively. The accuracies for top two categories for the classifier module were 98.32% and 98.52% on the training and test sets, respectively. This suggests that the planning layers (meta-layer and action-layer) are successfully revealing the most informative regions of the image.

Accuracy Vs. Epoch: In Fig. 9.5, we demonstrate the testing accuracy versus training epochs, which is based on hierarchical training sequence that was described in Subsection 9.3.1. The introduces two random baselines in addition to the final model: the model in which the goals and actions are decided in i.i.d. manner, in addition to the model in which the goals are planned, but the actions are planned in i.i.d. manner. Fig. 9.5 reveals that the errors in prediction have decreased by around 1/3 after using the goal planner, and by almost another 1/3 after incorporating the action planner.

Transfer Learning: In the previous experiment, all classification and planning layers were trained from scratch. However, transfer learning ideas suggest that we may accelerate

training if we can pretrain some modules. To this end, first, we consider ResNet-18 architecture and pretrain it on the the dataset (with full images) for 15 epochs. This resulted in more than 99% testing accuracy on the full images. Then, we replace the classification architecture in our system with ResNet-18 and start training *all layers* (i.e., planning and perception). The result of training is illustrated in Fig. 9.6, which shows that the maximum testing accuracy of 95.19% was achieved in a considerably shorter period of training (by almost an order of magnitude).

9.5 Concluding Remarks

We introduced a three-layer architecture for active perception of an image that allow us to co-design planning layers for goal generation and local navigation as well as classification layer. The layered structure of the proposed mechanism and the unified definition of reward for all layers enable us to train the parameters of the deep neural networks using a policy gradient algorithm. We would like to discuss a number of final remarks.

First, we did not use any overfitting prevention measures (dropouts, weight decay, etc.) in our models. However, even without use of validation sets, we observe a very good level of generalization of the current model. This may be explainable by use of fully-convolutional layers and global average pooling before evaluating the probability vectors, as suggested by [221].

Second, variations of the current architecture with recurrent memory (e.g., LSTM cells as used in [207]) are straight-forward to construct. This could be particularly useful when we extend our results to multi-robot scenarios.

Third, the intrinsic partial observability of this problem motivates use of policy gradient algorithms rather than Q-learning approaches [205]. It is an interesting line of research to develop Q-learning techniques that perform at the same level as the sampling based approaches for this class of problems.

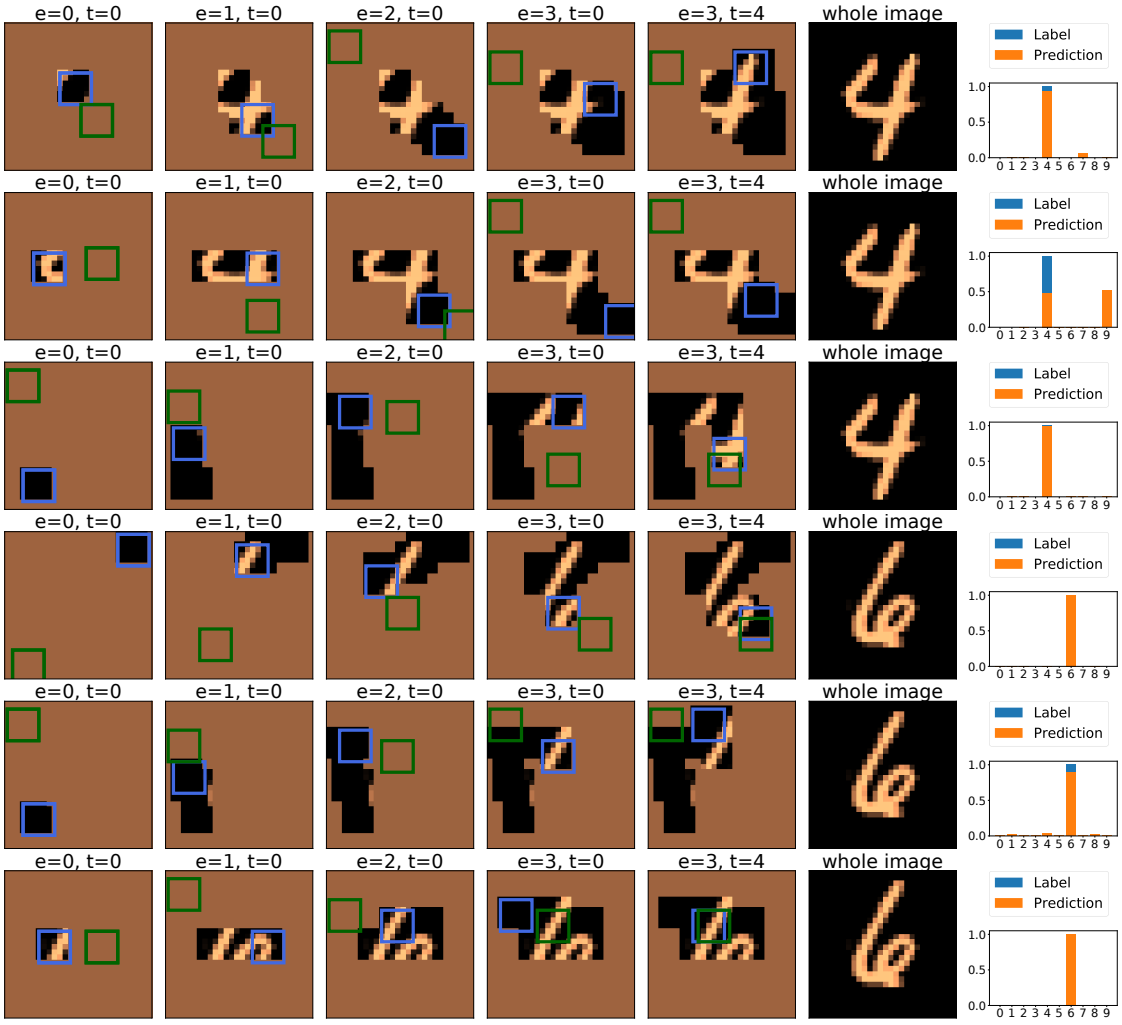

Figure 9.3: We demonstrate six sample trajectories from two data points. The snapshots have been taken at the beginning of 4 episodes as well as the end of the last episode. The blue and green squares point to the current position and goal position. The prediction corresponding to the final unmasked image is also illustrated in each case.

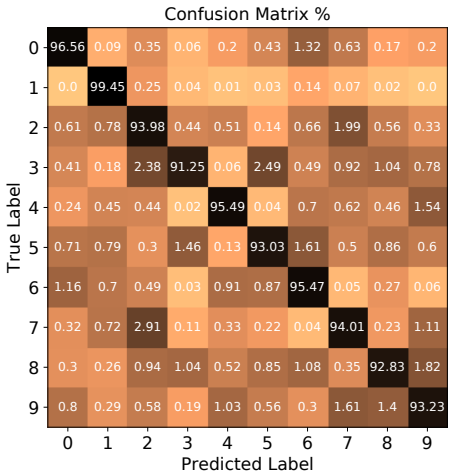

Figure 9.4: The confusion matrix of classification computed on the test dataset. The reported numbers are averaged over 20 runs of data.

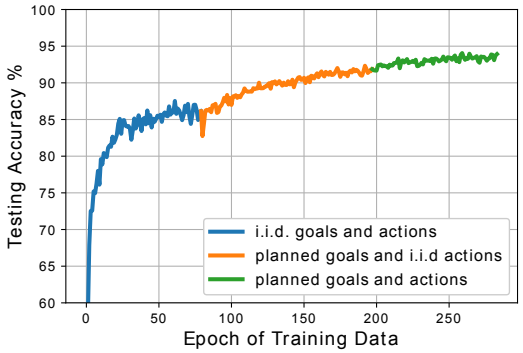

Figure 9.5: The testing data accuracy vs. data epoch using the hierarchical training sequence from scratch.

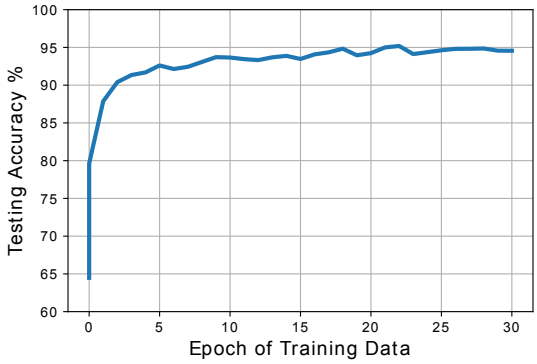

Figure 9.6: Testing data accuracy vs. data epoch with transfer learning.

Bibliography

[1] M. Siami and N. Motee, "Network abstraction with guaranteed performance bounds," *IEEE Transactions on Automatic Control*, 2018.

[2] M. Siami and N. Motee, "Growing linear dynamical networks endowed by spectral systemic performance measures," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2091–2106, 2017.

[3] A. Jadbabaie and A. Olshevsky, "Scaling laws for consensus protocols subject to noise," *IEEE Transactions on Automatic Control*, vol. 64, no. 4, pp. 1389–1402, 2018.

[4] M. Fardad, F. Lin, and M. R. Jovanović, "Design of optimal sparse interconnection graphs for synchronization of oscillator networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 9, pp. 2457–2462, 2014.

[5] M. Andreasson, E. Tegling, H. Sandberg, and K. H. Johansson, "Performance and scalability of voltage controllers in multi-terminal HVDC networks," in *2017 American Control Conference (ACC)*, pp. 3029–3034, IEEE, 2017.

[6] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[7] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, "Automated vehicle control developments in the PATH program," *IEEE Transactions on Vehicular Technology*, vol. 40, no. 1, pp. 114–130, 1991.

[8] R. Williams, B. Konev, and F. Coenen, "Multi-agent environment exploration with ar. drones," in *Conference Towards Autonomous Robotic Systems*, pp. 60–71, Springer, 2014.

[9] L. Merino, F. Caballero, J. R. Martínez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires," *Journal of Field Robotics*, vol. 23, no. 3-4, pp. 165–184, 2006.

[10] J. N. Tsitsiklis, *Problems in decentralized decision making and computation.* PhD thesis, Massachusetts Institute of Technology, 1984.

[11] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[12] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988–1001, 2003.

[13] R. Olfati-Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 2, pp. 951–956, IEEE, 2003.

[14] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[15] Y. Hatano and M. Mesbahi, "Agreement over random networks," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 2, pp. 2010–2015, IEEE, 2004.

[16] A. Tahbaz-Salehi and A. Jadbabaie, "A necessary and sufficient condition for consensus over random networks," *IEEE Transactions on Automatic Control*, vol. 53, no. 3, pp. 791–795, 2008.

[17] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.

[18] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, 2006.

[19] W. Ren and R. W. Beard, "Consensus algorithms for double-integrator dynamics," *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*, pp. 77–104, 2008.

[20] C. Nowzari and J. Cortés, "Self-triggered coordination of robotic networks for optimal deployment," *Automatica*, vol. 48, no. 6, pp. 1077–1087, 2012.

[21] B. Bamieh, M. R. Jovanovic, P. Mitra, and S. Patterson, "Coherence in large-scale networks: Dimension-dependent limitations of local feedback," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2235–2249, 2012.

[22] M. R. Jovanovic and B. Bamieh, "On the ill-posedness of certain vehicular platoon control problems," *IEEE Transactions on Automatic Control*, vol. 50, no. 9, pp. 1307–1321, 2005.

[23] R. H. Middleton and J. H. Braslavsky, "String instability in classes of linear time invariant formation control with limited communication range," *IEEE Transactions on Automatic Control*, vol. 55, no. 7, pp. 1519–1530, 2010.

[24] H. Hao and P. Barooah, "Stability and robustness of large platoons of vehicles with double-integrator models and nearest neighbor interaction," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 18, pp. 2097–2122, 2013.

[25] Y. Zheng, S. E. Li, K. Li, and W. Ren, "Platooning of connected vehicles with undirected topologies: Robustness analysis and distributed H-infinity controller synthesis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1353–1364, 2017.

[26] M. Siami and N. Motee, "Fundamental limits and tradeoffs on disturbance propagation in linear dynamical networks," *IEEE Transactions on Automatic Control*, vol. 61, no. 12, pp. 4055–4062, 2016.

[27] Z. Li, Z. Duan, and G. Chen, "On $\mathcal{H}_\infty$ and $\mathcal{H}_2$ performance regions of multi-agent systems," *Automatica*, vol. 47, no. 4, pp. 797–803, 2011.

[28] M. Pirani and S. Sundaram, "On the smallest eigenvalue of grounded Laplacian matrices," *IEEE Transactions on Automatic Control*, vol. 61, no. 2, pp. 509–514, 2015.

[29] Z. Li, X. Liu, P. Lin, and W. Ren, "Consensus of linear multi-agent systems with reduced-order observer-based protocols," *Systems & Control Letters*, vol. 60, no. 7, pp. 510–516, 2011.

[30] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[31] M. Chilali, P. Gahinet, and P. Apkarian, "Robust pole placement in LMI regions," *IEEE Transactions on Automatic Control*, vol. 44, no. 12, pp. 2257–2270, 1999.

[32] D. Famularo, P. Dorato, C. T. Abdallah, W. M. Haddad, and A. Jadbabaie, "Robust non-fragile LQ controllers: the static state feedback case," *International Journal of control*, vol. 73, no. 2, pp. 159–165, 2000.

[33] S. Weiland and J. C. Willems, "Almost disturbance decoupling with internal stability," *IEEE Transactions on Automatic Control*, vol. 34, no. 3, pp. 277–286, 1989.

[34] S. Patterson and B. Bamieh, "Leader selection for optimal network coherence," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, pp. 2692–2697, 2010.

[35] M. Gitterman, *The noisy oscillator: the first hundred years, from Einstein until now*. World Scientific, 2005.

[36] B. Schwartz, A. Isidori, and T. Tarn, "L_2 disturbance attenuation and performance bounds for linear non-minimum phase square invertible systems," in *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, vol. 1, pp. 227–228, IEEE, 1996.

[37] I. Gutman and A. Graovac, "Estrada index of cycles and paths," *Chemical Physics Letters*, vol. 436, no. 1, pp. 294–296, 2007.

[38] S. Boyd, "Lecture notes for EE263," *Introduction to Linear Dynamical Systems*, 2008.

[39] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, version 1.21 (2011)," *Available: cvxr. com/cvx*, 2010.

[40] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-space solutions to standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 831–847, 1989.

[41] J. P. Hespanha, "Lecture notes on LQR/LQG controller design," *Knowledge Creation Diffusion Utilization*, 2005.

[42] A. Saberi, Z. Lin, and A. A. Stoorvogel, "$\mathcal{H}_2$ almost disturbance decoupling problem with internal stability," in *American Control Conference, Proceedings of the 1995*, vol. 5, pp. 3414–3418, IEEE, 1995.

[43] B. Zhou, "On sum of powers of the Laplacian eigenvalues of graphs," *Linear Algebra and its Applications*, vol. 429, no. 8-9, pp. 2239–2246, 2008.

[44] A. E. Brouwer and W. H. Haemers, "A lower bound for the Laplacian eigenvalues of a graph—proof of a conjecture by guo," *Linear Algebra and its Applications*, vol. 429, no. 8-9, pp. 2131–2135, 2008.

[45] L. Qiu and E. J. Davison, "Performance limitations of non-minimum phase systems in the servomechanism problem," *Automatica*, vol. 29, no. 2, pp. 337–349, 1993.

[46] R. Dai and M. Mesbahi, "Optimal topology design for dynamic networks," in *2011 50th IEEE Conf. Decision Control / Euro. Control Conf.*, pp. 1280–1285, IEEE, 2011.

[47] F. Lin, M. Fardad, and M. R. Jovanović, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2426–2431, 2013.

[48] F. Lin, M. Fardad, and M. R. Jovanovic, "Optimal control of vehicular formations with nearest neighbor interactions," *IEEE Transactions on Automatic Control*, vol. 57, no. 9, pp. 2203–2218, 2012.

[49] N. Dhingra, F. Lin, M. Fardad, and M. R. Jovanovic, "On identifying sparse representations of consensus networks," 2012.

[50] M. Siami and N. Motee, "Network sparsification with guaranteed systemic performance measures," *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 246–251, 2015.

[51] D. A. Spielman and N. Srivastava, "Graph sparsification by effective resistances," *SIAM Journal on Computing*, vol. 40, no. 6, pp. 1913–1926, 2011.

[52] S. H. Moghaddam and M. R. Jovanović, "An interior point method for growing connected resistive networks," in *2015 American Control Conference*, pp. 1223–1228, 2015.

[53] M. H. de Badyn and M. Mesbahi, "Growing controllable networks via whiskering and submodular optimization," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pp. 867–872, IEEE, 2016.

[54] S. H. Moghaddam and M. R. Jovanović, "Customized algorithms for growing connected resistive networks," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 968–973, 2016.

[55] M. Siami and N. Motee, "Growing linear dynamical networks endowed by spectral systemic performance measures," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, 2018.

[56] S. Patterson, Y. Yi, and Z. Zhang, "A resistance distance-based approach for optimal leader selection in noisy consensus networks," *IEEE Transactions on Control of Network Systems*, 2018.

[57] J. Batson, D. A. Spielman, N. Srivastava, and S.-H. Teng, "Spectral sparsification of graphs: theory and algorithms," *Communications of the ACM*, vol. 56, no. 8, pp. 87–94, 2013.

[58] W. Ellens, F. Spieksma, P. Van Mieghem, A. Jamakovic, and R. Kooij, "Effective graph resistance," *Linear algebra and its applications*, vol. 435, no. 10, pp. 2491–2506, 2011.

[59] F. Dorfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 1, pp. 150–163, 2013.

[60] A. Beck, "The 2-coordinate descent method for solving double-sided simplex constrained minimization problems," *Journal of Optimization Theory and Applications*, vol. 162, no. 3, pp. 892–919, 2014.

[61] I. V. Konnov, "Selective bi-coordinate variations for resource allocation type problems," *Computational Optimization and Applications*, vol. 64, no. 3, pp. 821–842, 2016.

[62] A. S. Lewis, "Derivatives of spectral functions," *Mathematics of Operations Research*, vol. 21, no. 3, pp. 576–588, 1996.

[63] D. Miorandi, E. Altman, and G. Alfano, "The impact of channel randomness on coverage and connectivity of ad hoc and sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 3, pp. 1062–1072, 2008.

[64] I. Matei, J. S. Baras, and C. Somarakis, "Convergence results for the linear consensus problem under Markovian random graphs," *SIAM Journal on Control and Optimization*, vol. 51, pp. 1574–1591, 2013.

[65] J. Zhou and Q. Wang, "Convergence speed in distributed consensus over dynamically switching random networks," *Automatica*, vol. 45, no. 6, pp. 1455–1461, 2009.

[66] N. Abaid, I. Igel, and M. Porfiri, "On the consensus protocol of conspecific agents," *Lin. Algebra Applicat.*, vol. 437, no. 1, pp. 221–235, 2012.

[67] I. Matei, N. Martins, and J. S. Baras, "Almost sure convergence to consensus in Markovian random graphs," in *2008 47th IEEE Conference on Decision and Control*, pp. 3535–3540, IEEE, 2008.

[68] S. Martin, A. Girard, A. Fazeli, and A. Jadbabaie, "Multiagent flocking under general communication rule," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 2, pp. 155–166, 2014.

[69] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of parallel and distributed computing*, vol. 67, no. 1, pp. 33–46, 2007.

[70] C. F. Van Loan and N. Pitsianis, "Approximation with kronecker products," in *Linear algebra for large scale and real-time applications*, pp. 293–314, Springer, 1993.

[71] R. Yuster and U. Zwick, "Fast sparse matrix multiplication," *ACM Transactions On Algorithms (TALG)*, vol. 1, no. 1, pp. 2–13, 2005.

[72] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univ. Press, 2004.

[73] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted l-1 minimization," *Journal of Fourier analysis and applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

[74] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to derivative-free optimization*, vol. 8. Siam, 2009.

[75] L. El Ghaoui, "Inversion error, condition number, and approximate inverses of uncertain matrices," *Linear algebra and its applications*, pp. 171–193, 2002.

[76] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 6, pp. 777–790, 2001.

[77] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 3, pp. 2968–2973, IEEE, 2001.

[78] P. K. Wang, "Navigation strategies for multiple autonomous mobile robots moving in formation," *Journal of Field Robotics*, vol. 8, no. 2, pp. 177–195, 1991.

[79] C. Somarakis and J. S. Baras, "A simple proof of the continuous time linear consensus problem with applications in non-linear flocking networks," in *14th European Control Conference*, (Linz,Austria), July 2015.

[80] F. Dorfler and F. Bullo, "Synchronization and transient stability in power networks and nonuniform Kuramoto oscillators," *SIAM Journal on Control and Optimization*, vol. 50, no. 3, pp. 1616–1642, 2012.

[81] Y. Susuki and I. Mezic, "Nonlinear Koopman modes and a precursor to power system swing instabilities," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1182–1191, 2012.

[82] Q. Ali, N. Gageik, and S. Montenegro, "A review on distributed control of cooperating mini UAVs," *International Journal of Artificial Intelligence & Applications*, vol. 5, no. 4, p. 1, 2014.

[83] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, "Non-Bayesian social learning," *Games and Economic Behavior*, vol. 76, no. 1, pp. 210–225, 2012.

[84] F. Dörfler, M. Chertkov, and F. Bullo, "Synchronization in complex oscillator networks and smart grids," *Proceedings of the National Academy of Sciences*, vol. 110, no. 6, pp. 2005–2010, 2013.

[85] N. A. van Riel and E. D. Sontag, "Parameter estimation in models combining signal transduction and metabolic pathways: the dependent input approach," *IEE Proceedings-Systems Biology*, vol. 153, no. 4, pp. 263–274, 2006.

[86] G. Buzi, U. Topcu, and J. C. Doyle, "Quantitative nonlinear analysis of autocatalytic pathways with applications to glycolysis," in *American Control Conference (ACC), 2010*, pp. 3592–3597, IEEE, 2010.

[87] M. Siami and N. Motee, "On existence of hard limits in autocatalytic networks and their fundamental tradeoffs," *IFAC Proceedings Volumes*, vol. 45, no. 26, pp. 294–298, 2012.

[88] D. S. Bassett and E. Bullmore, "Small-world brain networks," *The neuroscientist*, vol. 12, no. 6, pp. 512–523, 2006.

[89] A. Jadbabaie, N. Motee, and M. Barahona, "On the stability of the Kuramoto model of coupled nonlinear oscillators," in *American Control Conference, Proc. 2004*, vol. 5, pp. 4296–4301, IEEE, 2004.

[90] A. Ajorlou, A. Momeni, and A. G. Aghdam, "Sufficient conditions for the convergence of a class of nonlinear distributed consensus algorithms," *Automatica*, vol. 47, no. 3, pp. 625–629, 2011.

[91] F. Cucker and S. Smale, "Emergent behavior in flocks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 852–862, 2007.

[92] H.-D. Chiang, F. F. Wu, and P. P. Varaiya, "A BCU method for direct analysis of power system transient stability," *IEEE Transactions on Power Systems*, vol. 9, no. 3, pp. 1194–1208, 1994.

[93] M. Arcak, "Passivity as a design tool for group coordination," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1380–1390, 2007.

[94] L. Shilong and S. Mao, "Aerodynamic force and flow structures of two airfoils in flapping motions," *Acta Mechanica Sinica*, vol. 17, no. 4, pp. 310–331, 2001.

[95] C. Somarakis and J. S. Baras, "Delay-independent convergence for linear consensus networks with applications to non-linear flocking systems," in *Submitted to the 12th IFAC Workshop on Time Delay Systems*, (Ann Arbor,MI,USA), June 2015.

[96] A. Papachristodoulou and A. Jadbabaie, "Synchonization in oscillator networks with heterogeneous delays, switching topologies and nonlinear dynamics," in *Decision and Control, 2006 45th IEEE Conference on*, pp. 4307–4312, IEEE, 2006.

[97] A. Papachristodoulou, A. Jadbabaie, and U. Munz, "Effects of delay in multi-agent consensus and oscillator synchronization," *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1471–1477, 2010.

[98] F. Cucker and E. Mordecki, "Flocking in noisy environments," *Journal de mathématiques pures et appliquées*, vol. 89, no. 3, pp. 278–296, 2008.

[99] Y. Kim and M. Mesbahi, "On maximizing the second smallest eigenvalue of a state-dependent graph Laplacian," in *Proceedings of the 2005, American Control Conference, 2005.*, pp. 99–103, IEEE, 2005.

[100] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[101] M. Siami and N. Motee, "Performance analysis of linear consensus networks with structured stochastic disturbance inputs," in *2015 American Control Conference*, pp. 4080–4085, 2015.

[102] H. K. Mousavi, C. Somarakis, and N. Motee, "Spectral performance analysis and design for distributed control of multi-agent systems," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, pp. 2918–2923, IEEE, 2017.

[103] H. K. Mousavi, C. Somarakis, M. Bahavarnia, and N. Motee, "Performance bounds and optimal design of randomly switching linear consensus networks," in *2017 American Control Conference (ACC)*, pp. 4347–4352, IEEE, 2017.

[104] C. Somarakis, Y. Ghaedsharaf, and N. Motee, "Time-delay origins of fundamental tradeoffs between risk of large fluctuations and network connectivity," *IEEE Transactions on Automatic Control*, 2019.

[105] A. Mauroy and I. Mezić, "Global stability analysis using the eigenfunctions of the Koopman operator," *IEEE Transactions on Automatic Control*, vol. 61, no. 11, pp. 3356–3369, 2016.

[106] Y. Lan and I. Mezić, "Linearization in the large of nonlinear systems and Koopman operator spectrum," *Physica D: Nonlinear Phenomena*, vol. 242, no. 1, pp. 42–53, 2013.

[107] M. Budišić, R. Mohr, and I. Mezić, "Applied Koopmanism," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 4, p. 047510, 2012.

[108] J. Kust, S. L. Brunton, B. W. Brunton, and P. J. L., *Dynamic Mode Decomposition. Data-Driven Modeling of Complex Systems*. OT149, SIAM, 2016.

[109] L. Perko, *Differential equations and dynamical systems*, vol. 7. Springer Science & Business Media, 2013.

[110] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.

[111] H. F. Cullen, *Introduction to General Topology*. D C Heath & Co, 1968.

[112] M.-M. Derriennic, "On multivariate approximation by bernstein-type polynomials," *Journal of Approximation Theory*, vol. 45, pp. 155–166, 1985.

[113] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[114] M. Abramowitz, I. A. Stegun, *et al.*, "Handbook of mathematical functions," *Applied mathematics series*, vol. 55, p. 62, 1966.

[115] C. A. Charalambides, *Enumerative combinatorics*. CRC Press, 2002.

[116] V. Barthelmann, E. Novak, and K. Ritter, "High dimensional polynomial interpolation on sparse grids," *Advances in Computational Mathematics*, vol. 12, no. 4, pp. 273–288, 2000.

[117] B. A. Malin, D. Krueger, and F. Kubler, "Solving the multi-country real business cycle model using a smolyak-collocation method," *Journal of Economic Dynamics and Control*, vol. 35, no. 2, pp. 229–239, 2011.

[118] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of basic engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[119] B. D. Anderson and J. B. Moore, "Optimal filtering," *Englewood Cliffs*, vol. 21, pp. 22–95, 1979.

[120] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[121] G. Bianchin, P. Frasca, A. Gasparri, and F. Pasqualetti, "The observability radius of networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3006–3013, 2017.

[122] S. Pequito, S. Kar, and A. P. Aguiar, "On the complexity of the constrained input selection problem for structural linear systems," *Automatica*, vol. 62, pp. 193–199, 2015.

[123] S. D. Pequito, S. Kar, A. P. Aguiar, *et al.*, "A framework for structural input/output and control configuration selection in large-scale systems.," *IEEE Trans. Automat. Contr.*, vol. 61, no. 2, pp. 303–318, 2016.

[124] A. Olshevsky, "Minimal controllability problems.," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.

[125] M. Egerstedt, S. Martini, M. Cao, K. Camlibel, and A. Bicchi, "Interacting with networks: How does structure relate to controllability in single-leader, consensus networks?," *IEEE Control Systems*, vol. 32, no. 4, pp. 66–73, 2012.

[126] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, "Sensing-constrained LQG control," in *2018 Annual American Control Conference (ACC)*, pp. 197–202, IEEE, 2018.

[127] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Near-optimal sensor scheduling for batch state estimation: Complexity, algorithms, and limits," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pp. 2695–2702, IEEE, 2016.

[128] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[129] A. Jadbabaie, A. Olshevsky, and M. Siami, "Deterministic and randomized actuator scheduling with guaranteed performance bounds," *arXiv preprint arXiv:1805.00606*, 2018.

[130] A. Jadbabaie, A. Olshevsky, and M. Siami, "Limitations and tradeoffs in minimum input selection problems," in *2018 Annual American Control Conference (ACC)*, pp. 185–190, IEEE, 2018.

[131] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.

[132] J. Wu, Q.-S. Jia, K. H. Johansson, and L. Shi, "Event-based sensor data scheduling: Trade-off between communication rate and estimation quality," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1041–1046, 2013.

[133] L. Shi, P. Cheng, and J. Chen, "Optimal periodic sensor scheduling with limited resources," *IEEE Transactions on Automatic Control*, vol. 56, no. 9, pp. 2190–2195, 2011.

[134] T. Tanaka and H. Sandberg, "SDP-based joint sensor and controller design for information-regularized optimal LQG control," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pp. 4486–4491, IEEE, 2015.

[135] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Transactions on signal processing*, vol. 62, no. 5, pp. 1135–1146, 2014.

[136] A. Marcus, D. A. Spielman, and N. Srivastava, "Interlacing families ii: Mixed characteristic polynomials and the kadison-singer problem," *arXiv preprint arXiv:1306.3969*, 2013.

[137] M. Brookes, "The matrix reference manual," *Imperial College London*, vol. 3, 2005.

[138] P. G. Casazza, G. Kutyniok, and F. Philipp, "Introduction to finite frame theory," in *Finite Frames*, pp. 1–53, Springer, 2013.

[139] V. K. Goyal, J. Kovačević, and J. A. Kelner, "Quantized frame expansions with erasures," *Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 203–233, 2001.

[140] M. Burrow, "The minimal polynomial of a linear transformation," *The American Mathematical Monthly*, vol. 80, no. 10, pp. 1129–1131, 1973.

[141] D. R. Karger, "Random sampling and greedy sparsification for matroid optimization problems," *Mathematical Programming*, vol. 82, no. 1-2, pp. 41–81, 1998.

[142] M. S. Bartlett, "An inverse matrix adjustment arising in discriminant analysis," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 107–111, 1951.

[143] A. Olshevsky, "On (non) supermodularity of average control energy," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1177–1181, 2018.

[144] R. H. Middleton, "Trade-offs in linear control system design," *Automatica*, vol. 27, no. 2, pp. 281–292, 1991.

[145] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback control theory*. Courier Corporation, 2013.

[146] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[147] L. Mirsky, "A trace inequality of john von Neumann," *Monatshefte für mathematik*, vol. 79, no. 4, pp. 303–306, 1975.

[148] C.-T. Chen, *Linear system theory and design*. Oxford University Press, Inc., 1998.

[149] B. Mityagin, "The zero set of a real analytic function," *arXiv preprint arXiv:1512.07276*, 2015.

[150] F. Hansen and G. K. Pedersen, "Jensen's operator inequality," *Bulletin of the London Mathematical Society*, vol. 35, no. 4, pp. 553–564, 2003.

[151] M. Fornasier and H. Rauhut, "Continuous frames, function spaces, and the discretization problem," *Journal of Fourier Analysis and Applications*, vol. 11, no. 3, pp. 245–287, 2005.

[152] S. Pequito, S. Kar, and A. P. Aguiar, "A framework for structural input/output and control configuration selection in large-scale systems," *IEEE Transactions on Automatic Control*, vol. 61, pp. 303–318, Feb 2016.

[153] C. Cheng, Y. Jiang, and Q. Sun, "Spatially distributed sampling and reconstruction of high-dimensional signals," in *Sampling Theory and Applications (SampTA), 2015 International Conference on*, pp. 453–457, IEEE, 2015.

[154] N. Motee and Q. Sun, "Sparsity measures for spatially decaying systems," in *American Control Conference (ACC), 2014*, pp. 5459–5464, IEEE, 2014.

[155] H. K. Mousavi and N. Motee, "Estimation with fast landmark selection in robot visual navigation," *arXiv preprint arXiv:1902.01026*, 2019.

[156] H. K. Mousavi, Q. Sun, and N. Motee, "Space-time sampling for network observability," *arXiv preprint arXiv:1811.01303*, 2018.

[157] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[158] F. Hansen, J. Pečarić, and I. Perić, "Jensen's operator inequality and its converses," *Mathematica Scandinavica*, pp. 61–73, 2007.

[159] G.-Z. Yang, J. Bellingham, P. E. Dupont, P. Fischer, L. Floridi, R. Full, N. Jacobstein, V. Kumar, M. McNutt, R. Merrifield, *et al.*, "The grand challenges of science robotics," *Science Robotics*, vol. 3, no. 14, p. eaar7650, 2018.

[160] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[161] R. Sim and G. Dudek, "Learning and evaluating visual features for pose estimation," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1217–1222, IEEE, 1999.

[162] P. Sala, R. Sim, A. Shokoufandeh, and S. Dickinson, "Landmark selection for vision-based navigation," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 334–349, 2006.

[163] R. Lerner, E. Rivlin, and I. Shimshoni, "Landmark selection for task-oriented navigation," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 494–505, 2007.

[164] S. Chen, Y. F. Li, W. Wang, and J. Zhang, *Active sensor planning for multiview vision tasks*, vol. 1. Springer, 2008.

[165] H. Strasdat, C. Stachniss, and W. Burgard, "Which landmark is useful? learning selection policies for navigation in unknown environments," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 1410–1415, IEEE, 2009.

[166] A. Gorbenko and V. Popov, "The problem of selection of a minimal set of visual landmarks," *Applied Mathematical Sciences*, vol. 6, no. 95, pp. 4729–4732, 2012.

[167] M. Beinhofer, J. Müller, A. Krause, and W. Burgard, "Robust landmark selection for mobile robot navigation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3637–2643, IEEE, 2013.

[168] B. Mu, L. Paull, A.-A. Agha-Mohammadi, J. J. Leonard, and J. P. How, "Two-stage focused inference for resource-constrained minimal collision navigation," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 124–140, 2017.

[169] A. J. Davison, "Active search for real-time vision," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 66–73, IEEE, 2005.

[170] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 1–20, 2018.

[171] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[172] F. Lin, M. Fardad, and M. R. Jovanović, "Algorithms for leader selection in stochastically forced consensus networks," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1789–1802, 2014.

[173] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Sensor placement for optimal Kalman filtering: Fundamental limits, submodularity, and algorithms," in *American Control Conference (ACC), 2016*, pp. 191–196, IEEE, 2016.

[174] M. A. K. Bahrin, M. F. Othman, N. H. N. Azli, and M. F. Talib, "Industry 4.0: A review on industrial automation and robotic," *Jurnal Teknologi*, vol. 78, no. 6-13, 2016.

[175] L. Yushi, J. Fei, and Y. Hui, "Study on application modes of military internet of things (miot)," in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 3, pp. 630–634, IEEE, 2012.

[176] C. Bhatt, N. Dey, and A. S. Ashour, *Internet of things and big data technologies for next generation healthcare*, vol. 23. Springer, 2017.

[177] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems*, pp. 6379–6390, 2017.

[178] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," *arXiv preprint arXiv:1802.08757*, 2018.

[179] A. Khan, C. Zhang, V. Kumar, and A. Ribeiro, "Collaborative multiagent reinforcement learning in homogeneous swarms," 2018.

[180] D. Silver, L. Newnham, D. Barker, S. Weller, and J. McFall, "Concurrent reinforcement learning from customer interactions," in *International Conference on Machine Learning*, pp. 924–932, 2013.

[181] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, *et al.*, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[182] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, ACM, 2017.

[183] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan, *et al.*, "Towards federated learning at scale: System design," *arXiv preprint arXiv:1902.01046*, 2019.

[184] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, pp. 2137–2145, 2016.

[185] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[186] T. Nguyen and S. Mukhopadhyay, "Selectively decentralized Q-learning," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 328–333, IEEE, 2017.

[187] M. Fazel, R. Ge, S. M. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," *arXiv preprint arXiv:1801.05039*, 2018.

[188] S. Alemzadeh and M. Mesbahi, "Distributed Q-learning for dynamically decoupled systems," *arXiv preprint arXiv:1809.08745*, 2018.

[189] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[190] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[191] A. D. Domínguez-García and C. N. Hadjicostis, "Distributed strategies for average consensus in directed graphs," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 2124–2129, IEEE, 2011.

[192] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, pp. 1057–1063, 2000.

[193] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[194] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.

[195] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[196] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in neural information processing systems*, pp. 315–323, 2013.

[197] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 528–535, IEEE, 2016.

[198] C. Connolly, "The determination of next best views," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 432–435, IEEE, 1985.

[199] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning $so(3)$ equivariant representations with spherical CNNs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 52–68, 2018.

[200] D. W. Aha, "Goal reasoning: Foundations, emerging applications, and prospects.," *AI Magazine*, vol. 39, no. 2, pp. 3–24, 2018.

[201] H. Muñoz-Avila, "Adaptive goal driven autonomy," in *International Conference on Case-Based Reasoning*, pp. 3–12, Springer, 2018.

[202] M. A. Wilson, J. McMahon, A. Wolek, D. W. Aha, and B. H. Houston, "Goal reasoning for autonomous underwater vehicles: Responding to unexpected agents," *AI Communications*, vol. 31, no. 2, pp. 151–166, 2018.

[203] D. Dannenhauer and H. Muñoz-Avila, "Goal-driven autonomy with semantically-annotated hierarchical cases," in *International Conference on Case-Based Reasoning*, pp. 88–103, Springer, 2015.

[204] M. W. Floyd, J. Karneeb, P. Moore, and D. W. Aha, "A goal reasoning agent for controlling UAVs in beyond-visual-range air combat.," in *IJCAI*, pp. 4714–4721, 2017.

[205] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," in *Advances in neural information processing systems*, pp. 3675–3683, 2016.

[206] K. Erol, *Hierarchical task network planning: formalization, analysis, and implementation.* PhD thesis, 1996.

[207] H. K. Mousavi, M. Nazari, M. Takáč, and N. Motee, "Multi-agent image classification via reinforcement learning," *arXiv preprint arXiv:1905.04835*, 2019.

[208] S. D. Whitehead and D. H. Ballard, "Active perception and reinforcement learning," in *Machine Learning Proceedings 1990*, pp. 179–188, Elsevier, 1990.

[209] Y. Aloimonos, *Active perception.* Psychology Press, 2013.

[210] D. H. Ballard, M. M. Hayhoe, F. Li, and S. D. Whitehead, "Hand-eye coordination during sequential tasks," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 337, no. 1281, pp. 331–339, 1992.

[211] J. K. Tsotsos, *A computational perspective on visual attention*. MIT Press, 2011.

[212] M. Balcarras, S. Ardid, D. Kaping, S. Everling, and T. Womelsdorf, "Attentional selection can be predicted by reinforcement learning of task-relevant stimulus features weighted by value-independent stickiness," *Journal of Cognitive Neuroscience*, vol. 28, no. 2, pp. 333–349, 2016.

[213] R. G. Mesquita and C. A. Mello, "Object recognition using saliency guided searching," *Integrated Computer-Aided Engineering*, vol. 23, no. 4, pp. 385–400, 2016.

[214] N. D. Bruce, C. Wloka, N. Frosst, S. Rahman, and J. K. Tsotsos, "On computational modeling of visual saliency: Examining what's right, and what's left," *Vision research*, vol. 116, pp. 95–112, 2015.

[215] A. Bruno, F. Gugliuzza, E. Ardizzone, C. C. Giunta, and R. Pirrone, "Image content enhancement through salient regions segmentation for people with color vision deficiencies," *i-Perception*, vol. 10, no. 3, p. 2041669519841073, 2019.

[216] E. Potapova, M. Zillich, and M. Vincze, "Survey of recent advances in 3D visual attention for robotics," *The International Journal of Robotics Research*, vol. 36, no. 11, pp. 1159–1176, 2017.

[217] B. Schauerte, "Bottom-up audio-visual attention for scene exploration," in *Multimodal Computational Attention for Scene Understanding and Robotics*, pp. 35–113, Springer, 2016.

[218] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[219] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[220] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[221] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

Vita

Hossein K. Mousavi (Seyyedhossein Mousavi) received his B.Sc. degree in Mechanical Engineering from Sharif University of Technology, Tehran, Iran in 2014. From 2015 to 2019, he obtained Ph.D. in Mechanical Engineering from Lehigh University, Bethlehem, Pennsylvania. His current line of research is optimal decision-making and estimation in networked control systems, machine learning, and robotics. He is currently a Machine Learning Research Engineer at Apple Inc..